Gestion des fonctions réseau et du trafic dans les réseaux virtualisés

par

Tarik MOUFAKIR

THÈSE PRÉSENTÉE À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION
DU DOCTORAT EN GÉNIE
Ph.D.

MONTRÉAL, LE "24 JANVIER 2022"

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CETTE THÈSE A ÉTÉ ÉVALUÉE

PAR UN JURY COMPOSÉ DE:

M. Mohamed Faten Zhani, Directeur de thèse
Département de génie logiciel et des TI à l'École de Technologie Supérieure

M. Abdelouahed Gherbi, Co-directeur
Département de génie logiciel et des TI à l'École de Technologie Supérieure

M. Georges Kaddoum, Président du jury
Département de génie électrique à l'École de Technologie Supérieure

M. Chamseddine Talhi, Membre du jury
Département de génie logiciel et des TI à l'École de Technologie Supérieure

M. Wessam Ajib, Examinateur externe indépendant
Département d'informatique à Université du Québec à Montréal

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

LE "25 NOVEMBRE 2021"

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

AVANT-PROPOS

Dans cette thèse de doctorat en Technologies de l'Information et des Communications (TIC), nos travaux de recherches visent à aborder des défis liés à la gestion des réseaux virtualisés, notamment l'allocation des ressources pour les chaines et les fonctions réseaux ainsi que le routage du trafic dans un seul domaine ou à travers des domaines multiples. Ainsi, on s'intéresse à deux volets :

- le gestion des chaines de service réseau en tant que service (*Service Function Chain as a Service* - SFCaaS) dans les réseaux virtualisés,
- la gestion de trafic dans les réseaux multi-domaines.

REMERCIEMENTS

Tout d'abord, je souhaite remercier mon directeur de recherche, le professeur Mohamed Faten Zhani. Je le remercie d'avoir cru en moi et de m'avoir soutenu tout au long de la réalisation de ce travail. Ses conseils m'ont été d'une grande valeur. Ses compétences, ses qualités humaines et professionnelles m'ont profondément marqué.

Je voudrais également remercier le professeur Abdelouahed Gherbi pour son aide inestimable. Je le remercie pour les connaissances qu'il m'a inculquées, les discussions scientifiques enrichissantes et le temps qu'il m'a consacré.

J'exprime mes sincères remerciements à mes parents, mon frère, mes sœurs, mes amis et à toute ma famille pour leur soutien et leur amour tout au long de mes études.

Enfin, je voudrais remercier tout particulièrement ma conjointe et mes deux enfants pour leur patience et leurs sacrifices. Ils étaient toujours présents pour me remonter le moral et me soutenir. Ils m'ont toujours soutenu et encouragé énormément afin de réaliser avec succès cette thèse.

Je désire aussi remercier celles et ceux qui ont participé de près ou de loin à l'élaboration de ce travail.

Gestion des fonctions réseau et du trafic dans les réseaux virtualisés

Tarik MOUFAKIR

RÉSUMÉ

La virtualisation des fonctions réseau (*Network Function Virtualization* - NFV) et la réseautique définie par logiciel (*Software Defined Networking* - SDN) sont deux nouveaux paradigmes qui ont été récemment introduits qui permettent de changer la façon avec laquelle les réseaux sont configurés et maintenus. NFV et SDN offrent plusieurs avantages, notamment la création et la reconfiguration dynamique des VNFs ainsi que le routage dynamique du trafic.

Dans ce contexte, nous nous intéressons, dans la première partie de cette thèse, à la possibilité d'offrir les chaînes de fonctions de service (*Service Function Chain* - SFC) en tant que service (SFCaaS) où une chaîne de service pourrait être proposée en tant que service à un tiers. Un SFC est composé d'un ensemble de fonctions de réseau virtuelles (*Virtual Network Functions* - VNFs) qui sont implémentées dans une machine virtuelle ou conteneur s'exécutant sur un serveur ou un équipement dédié et qui liés par des liens virtuels pour transporter le trafic. Ainsi, nous cherchons à résoudre le problème d'allocation de ressources aux SFCs dans l'infrastructure physique et le routage de leur trafic. Nous formulons donc le problème en tant qu'un programme linéaire en nombres entiers (*Integer Linear Program* - ILP) et proposons un algorithme heuristique visant à maximiser le revenu total du fournisseur de SFC en tenant compte du coût des instances, le coût d'exploitation opérationnelle, et le coût de synchronisation entre les instances VNF.

Dans la deuxième partie de cette thèse, nous nous intéressions à la gestion du trafic dans les réseaux multi-domaines où chaque domaine est administré et géré par un seul opérateur de réseau. Malheureusement, généralement, les opérateurs réseau ne collaborent pas pour prendre leurs décisions de routage et les performances globales du réseau multi-domaine. Motivés par la nécessité de résoudre ce problème, nous proposons un nouveau mécanisme de routage collaboratif multi-domaines capable de router efficacement les flux entrants à travers plusieurs domaines tout en garantissant leurs exigences de performance en termes de délai et de bande passante et en maximisant l'utilisation globale du réseau. Nous proposons donc un programme linéaire entier pour résoudre ce problème et développons un algorithme heuristique adapté aux grandes échelles. Les résultats des simulations montrent que le mécanisme proposé est capable d'optimiser considérablement l'utilisation du réseau et de maximiser le nombre de flux routés avec des performances garanties.

Mots-clés: réseaux définis par logiciel, routage multi-domaine, performance du réseau, chaînes de fonctions de service en tant que service (SFCaaS), fonction de réseau virtuelle (VNF), chaîne de fonctions de service (SFC)

Managing network functions and traffic in virtualized networks

Tarik MOUFAKIR

ABSTRACT

Network Function Virtualization (NFV) and Software Defined Networking (SDN) are two new paradigms that have recently been introduced that are changing the way networks are configured and maintained. NFV and SDN offer several advantages, including dynamic creation and reconfiguration of VNFs as well as dynamic traffic routing.

In this context, we are interested, in the first part of this thesis, in the possibility of offering Service Function Chain (SFC) as a service (SFCaaS) where a chain of service could be offered as a service to a third party. An SFC is made up of a set of virtual network functions (VNFs) which are implemented in a virtual machine or container running on a dedicated server or device and which are linked by virtual links to carry traffic. Thus, we try to solve the problem of allocating resources to SFCs in the physical infrastructure and the routing of their traffic. We therefore formulate the problem as an Integer Linear Program (ILP) and propose a heuristic algorithm aimed at maximizing the total revenue of the SFC provider taking into account the cost of instances, the operating cost operational, and the cost of synchronization between VNF instances.

In the second part of this thesis, we are interested in traffic management in multi-domain networks where each domain is administered and managed by a single network operator. Unfortunately, network operators typically do not work together to make their routing decisions and the overall performance of the multi-domain network. Motivated by the need to solve this problem, we propose a new multi-domain collaborative routing mechanism capable of efficiently routing inbound flows across multiple domains while ensuring their performance requirements in terms of delay and bandwidth and maximizing l overall use of the network. We therefore propose an entire linear program to solve this problem and develop a heuristic algorithm suitable for large scales. The results of the simulations show that the proposed mechanism is capable of considerably optimizing the use of the network and of maximizing the number of flows routed with guaranteed performance.

Keywords: software defined networking, multi-domain routing, network performance, service function chains as a service (SFCaaS), virtualized network function (VNF), service function chain (SFC)

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ALGORITHMES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

API	Application Programming Interface
APIaaS	API as a service
ARPANET	Advanced Research Projects Agency Network
AS	Autonomous systems
BaaS	Backend as a service
BGP	Border Gateway Protocol
BGP-LS	Border Gateway Protocol-Link State
BRPC	Backward Recursive PCE-based Computation
CAGR	Compound Annual Growth Rate
CapEx	CApital EXPenditure
CDF	Cumulative Distribution Function
cPCEs	children PCEs
CPU	Central Processing Unit
CRM	Customer Relationship Management
DaaS	Data as a service / Desktop as a service
DARPA	Defense Advanced Research Projects Agency
DBaaS	Database as a service
DPI	Database as a service
EC2	Cloud Elastic Compute
ETS	École de Technologie Supérieure
ETSI	European Telecommunications Standards Institute
FPGAs	Field-Programmable Gate Arrays
GMPLS	Generalized Multi-Protocol Label Switching

H-PCE	HierarchicalPath Computation Element
IaaS	Infrastructure as a Service
ICT	Information and communications technology
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
ILP	Intger Linear Programming
IP	Internet Protocol
IT	Information technology
IXP	Internet eXchange Points
MIT	Massachusetts Institute of Technology
MPLS	Multi-Protocol Label Switching
NAS	Network Attached Storage
NAT	Network Address Translation
NFC	Network Function Chains
NFCaaS	Network Function Chains as a Service
NF	Network Function
NFV	Network Function Virtualization
NIST	National Institute of Standards and Technology
ONF	Open Networking Foundation
OpEx	Operating Expenditure
OSI	Open Systems Interconnection
OQLF	Office québécois de la langue française
OSPF	Open Shortest Path First
OVS	Open VSwitch

PaaS	Platform as a Service
PCE	Path Computation Element
PCEP	PCE communication protocol
PoP	Point of Presence
pPCE	parent PCE
QoS	Quality of Service
RH	Ressources Humaines
ROI	Return On Investment
S3	Simple Storage Service
SaaS	Software as a Service
SAN	Storage Area Network
SECaaS	Security as a service
SFCaaS	Service Function Chain as a service
SDN	Software Defined Network
SDX	software defined Internet exchange
SFC	Service Function Chain
sFlow	sampled Flow
SLA	Service Level Agreement
SP	Service Provider
SPOF	Single point of failure
STaaS	Storage as a service
STP	Spanning Tree Protocol
TCAM	Ternary Content Addressable Memory
TCO	Total cost of ownership (coût total de possession)
TCP	Transmission Control Protocol

TEaaS	Test environment as a service
UDP	User Datagram Protocol
VMs	Virtual Machines
VNF	Virtualized Network Function
VSPT	Virtual Shortest Path Tree

LISTE DES SYMBOLES ET UNITÉS DE MESURE

GB	Gigabyte (1,000,000,000 bytes)
GiB	Gibibyte (1 GiB = 1.073741824 GB)
Gbps	Giga Bit par seconde
GHz	Giga Hertz
k-pkt	kilo packet
msec	Milliseconde
min	Minutes
pps	Paquets par seconde
rps	requêtes par seconde
sec	Second

INTRODUCTION

0.1 Contexte général

L'infonuagique (*Cloud Computing*), la virtualisation des fonctions réseau (*Network Function Virtualization* - NFV) et la réseautique définie par logiciel (*Software Defined Networking* - SDN) sont de nouveaux paradigmes qui ont récemment émergé et qui permettent de changer la façon avec laquelle les infrastructures et les réseaux informatiques sont approvisionnés, configurés et maintenus. Ces technologies offrent plusieurs avantages, notamment d'accéder rapidement et dynamiquement à d'énormes ressources informatiques et de pouvoir créer et gérer des fonctions réseau virtuelles (*Virtual Network Functions* - VNFs) à la demande et de router le trafic d'une façon dynamique. Elles offrent aussi l'opportunité de réaliser des économies importantes, puisqu'elles favorisent une meilleure utilisation des ressources à moindre coût offerte selon les besoins des clients.

Avec la maturité de ces technologies, plusieurs propositions telles que FlexNGIA Zhani & Elbakoury (2020) ont été faites afin de revisiter l'architecture et les protocoles de l'internet d'aujourd'hui. En effet, FlexNGIA préconise d'exploiter les technologies de Cloud Computing, NFV et SDN afin de révolutionner l'architecture de l'Internet en permettant d'offrir des chaînes de fonctions de service (*Service Function Chain* - SFC) en tant que service (SFCaaS). Un SFC est composé d'un ensemble de fonctions de réseau virtuel (VNF) (telles que les pare-feu et les systèmes de détection d'intrusion - IDS), qui sont implémentées dans une machine virtuelle ou conteneur s'exécutant sur un serveur ou un équipement dédié et qui sont liés par des liens virtuels pour transporter le trafic. Ces fonctions réseau sont connectées à travers des liens virtuels garantissant leur bande passante et leur délai de propagation.

Offrir les SFCs en tant que service permet non seulement de personnaliser les fonctions réseau et de les approvisionner à la demande, mais aussi de personnaliser les protocoles réseau qui

gèrent la communication entre les composantes de la chaîne. Cela représente une révolution puisque FlexNGIA n'offre pas seulement la connectivité, mais aussi un ensemble de fonctions et de protocoles dédiés à chaque application contrairement à l'internet d'aujourd'hui qui se base principalement sur les algorithmes de routages standards et offre seulement la connectivité entre la source et la destination des données.

La mise en place de cette nouvelle architecture nécessite la résolution de plusieurs défis de recherche tels que la conception de chaînes de service et de fonction réseaux personnalisées, la mise en place de nouveaux protocoles de transport plus adaptés aux applications réseau du futur, l'allocation dynamique des ressources pour les chaînes de services afin de réduire les coûts opérationnels tout en garantissant la performance, la surveillance et le monitorage des applications et de l'infrastructure et aussi la gestion des ressources et du trafic dans le cas où plusieurs domaines (c.-à-d. des réseaux séparés qui sont gérés par des administrateurs indépendants) sont interconnectés.

Dans ce travail, nous nous attaquons particulièrement à deux défis parmi ceux qui sont cités, à savoir, l'allocation de ressources pour les SFCs et la gestion du trafic dans les réseaux multi-domaines. Plus de détails sur ces deux défis sont présentés dans le paragraphe suivant.

0.2 Problématique et objectifs

Dans cette thèse, nous nous visons à réaliser les objectifs suivants :

- l'allocation des ressources pour les SFCs,
- et la gestion efficace de trafic dans les réseaux multi-domaines.

Dans ce qui suit, nous donnons plus détails sur ces deux objectifs.

0.2.1 Allocation des ressources aux SFCs

Les avancées technologiques de virtualisation, la virtualisation des fonctions réseau (NFV), la réseautique définie par logiciel (SDN) et la programmabilité du plan de données du réseau ont conduit à l'apparition d'une nouvelle tendance appelée 'softwarization' qui vise à implémenter les fonctions réseau en logiciel (*software*) au lieu de les avoir dans des équipements dédiés. Cela permettrait d'offrir des chaînes de fonction de services en tant que service (SFCaaS) où une chaîne de fonction de service (SFC) pourrait être proposée en tant que service à un tiers. Les VNFs constituants la chaîne peuvent être des fonctions de réseau (par exemple, pare-feu, IDS) implémentées dans une machine virtuelle ou un conteneur s'exécutant sur un serveur de base ou via un matériel programmable (par exemple, les cartes *Field-Programmable Gate Arrays* - FPGA).

Dans ce contexte, nous nous intéressons à l'un des principaux défis des fournisseurs de SFCs qui consiste à allouer les ressources pour les SFCs tout en maximisant les profits, minimisant les coûts opérationnels (coûts des instances, des liens virtuels et de la transmission de trafic) et en satisfaisant les exigences des SFCs en termes de délais de bout en bout, taux de perte et bande passante. Pour satisfaire ces besoins, les fournisseurs devront répondre à plusieurs questions : comment identifier le nombre optimal d'instances (c-à-d., machines virtuelles) et leurs types pour implémenter chaque VNF constituant la chaîne ? comment placer les instances dans le réseau physique tout en tenant compte du coût de déploiement (qui varie considérablement d'un emplacement à l'autre), et aussi chaîner les instances et garantir la performance requise pour la chaîne ?

0.2.2 Gestion du trafic dans les réseaux multi-domaines

La réseautique définie par logiciel (*Software Defined Networking* – SDN) offre plus de flexibilité et d'agilité aux opérateurs réseau pour configurer dynamiquement les équipements du réseau et

les adapter le routage du trafic selon les besoins des utilisateurs. Ce nouveau paradigme propose une nouvelle architecture réseau dans laquelle le plan de contrôle, qui est responsable de la prise des décisions de routage, est séparé du plan de données, qui est responsable du transfert des paquets. Ainsi, un contrôleur central est utilisé pour gérer et contrôler le réseau d'une manière automatique et dynamique. Il est donc capable de prendre les meilleures décisions de routage en se basant sur une vue globale du réseau SDN.

Malgré les bienfaits attendus de cette technologie, il y a encore plusieurs défis reliés à la performance dans le contexte des échanges de données entre des domaines multiples. En effet, les paquets de données peuvent traverser plusieurs domaines SDN gérés par différents opérateurs réseau. Cependant, sans collaboration entre les opérateurs et sans possibilité de partager des informations internes sur les caractéristiques et l'état de leurs réseaux, chaque domaine est vu comme une "boîte noire" par les autres.

L'absence de coopération entre les différents domaines, qui participent à la transmission des paquets de données, entraîne de nombreuses limitations :

- il n'est pas possible de déterminer le chemin optimal de bout en bout traversant les différents domaines entre la source et la destination,
- il n'est pas possible de connaître l'état des chemins dans les différents domaines (par exemple, l'utilisation du réseau, le délai de transmission, la bande passante),
- il n'est pas possible de connaître à l'avance les différents chemins possibles pour atteindre la destination au niveau de chaque domaine,
- et finalement, il ne sera pas possible de garantir la performance requise en termes de délai, débit et taux de perte pour les applications pour lesquelles les paquets doivent traverser plusieurs domaines administratifs.

Le partage d'informations internes sur les caractéristiques et l'état entre différents domaines permettra de prendre les meilleures décisions de routage. Une telle collaboration pourra

augmenter considérablement les performances du routage multi-domaines et offrir une meilleure utilisation du réseau et réduire le risque de congestion.

0.3 Contributions

Lors de cette thèse, nous avons apporté des contributions dans les deux volets suivants :

- l'allocation des ressources réseaux à travers le nuage comme service,
- et la gestion de trafic dans les réseaux multi-domaines.

Nos contributions peuvent être décrites comme suit :

- Volet : Allocation des ressources aux SFCs
  - Nous avons décomposé le problème d'approvisionnement de ressource pour un SFC en deux phases. La première phase est celle de traduction qui vise à identifier le nombre optimal d'instances (c.-à-d., machines virtuelles) qui sont nécessaires pour chaque VNF afin de répondre à la demande (c.-à-d., le trafic entrant). La deuxième phase est la phase de "mappage" visant à décider où placer les instances en tenant compte du coût de déploiement, qui varie considérablement d'un emplacement à l'autre. À notre connaissance, aucun travail préalable n'a considéré la phase de traduction.
  - Nous avons effectué une étude détaillée des coûts des machines virtuelles (instances) offertes par Amazon EC2 (Amazon) afin d'analyser l'impact de l'emplacement, la taille de l'instance sur le coût et la performance de l'instance. Cette étude étend celle réalisée par (Ghrada, Zhani & Elkhatib, 2018).
  - Nous formulons de problème de la phase de mappage en tant qu'un programme linéaire entier visant à réduire les coûts opérationnels des instances et des liens du fournisseur SFC ainsi que les coûts de synchronisation entre les instances de même type.
  - Nous proposons deux algorithmes heuristiques pour résoudre le problème de mappage avec les mêmes objectifs mentionnés précédemment en tenant compte de l'étude des

coûts des instances Amazon EC2. Le premier est un algorithme de base intuitif (appelé : ‘Baseline‘) et le second est un algorithme plus sophistiqué de Provisioning basé sur la composition SFC (appelé : ‘SPIN‘) qui fournit des résultats considérablement améliorés par rapport au Baseline.

- Volet : Gestion du trafic dans les réseaux multi-domaines
  - Nous proposons un nouveau mécanisme de routage multi-domaines collaboratif qui tire profit de la collaboration entre les opérateurs de plusieurs domaines afin d’acheminer efficacement les flux entre plusieurs domaines tout en satisfaisant leurs performances de bout en bout et leurs exigences de coût.
  - Nous formulons le problème en tant qu’un programme linéaire entier (*Integer Linear Program* - IPL).
  - Nous avons conçu et développé un algorithme heuristique pour maximiser le nombre des flux transmis avec succès pour les réseaux à grandes échelles.
  - Nous avons démontré l’efficacité du mécanisme de routage multi-domaines collaboratif pour augmenter considérablement le nombre de flux transmis en respectant exigences de performances et de coûts.

0.4 Structure de la thèse

La suite de cette thèse est organisée comme suit. Dans le chapitre 1, nous présentons les concepts de base reliés à l’infonuagique ainsi que les technologies qui y sont connexes. Nous présentons d’abord les concepts de l’infonuage, ensuite, nous présentons les concepts de la virtualisation, la réseautique définie par logiciel, ainsi que la virtualisation des fonctions réseau. Enfin, nous terminons ce chapitre par une présentation des chaînes de service de réseau et les concepts de l’ordonnancement.

Le chapitre 2 présente une revue de la littérature sur le sujet mettant en évidence la nouveauté de ce travail.

Dans le chapitre 3, nous exposons notre contribution au niveau de l'allocation des ressources réseaux à travers le nuage comme service.

Le chapitre 4 vise à montrer notre contribution pour améliorer la gestion du trafic dans les réseaux multi-domaines.

Le dernier chapitre résume cette thèse par une conclusion qui exprime les contributions réalisées dans le cadre de nos recherches, ainsi que des perspectives qui peuvent faire l'objet de futurs travaux de recherche.

CHAPITRE 1

L'INFONUAGIQUE ET LES TECHNOLOGIES CONNEXES

1.1 Introduction

Dans ce chapitre, nous présentons les concepts de base reliés à l'infonuagique ainsi que les technologies qui y sont connexes, plus particulièrement, la réseautique définie par logiciel et la virtualisation. Nous décrivons aussi les notions reliées à la virtualisation des fonctions réseau et le chaînage de fonctions réseau.

1.2 L'infonuagique

1.2.1 Définitions et terminologie

L'infonuagique (*Cloud Computing*) se démarque par sa popularité en tant qu'un modèle économique qui a suscité beaucoup d'intérêt au cours des dernières années et qui a révolutionné le monde de l'informatique (Jeba, Roy, Rashid, Atik & Whaiduzzaman, 2019). Aujourd'hui, l'infonuagique est la solution permettant répondre aux besoins informatiques (par exemple, applications, stockage, etc) des organisations et des individus sans se préoccuper de la gestion des infrasturcutres technologiques sous-jacentes.

L'infonuagique ou *cloud computing* est un terme anglophone qui a fait son apparition à la fin des années 1990 aux États-Unis. La communauté francophones canadiens ont adopté le terme « Infonuagique » comme traduction de l'expression "*Cloud computing*", tandis qu'en France La Commission Générale de Terminologie et de Néologie a choisi l'expression « Informatique en nuage » (Lacaze, 2013).

Actuellement, les termes "infonuagique" ou le "*Cloud computing*" sont employés parfois partout et ils ont été souvent malmenés par tout le grand public et même par les professionnels des technologies de l'information (TI). Vaquero et al ont. identifié plus d'une vingtaine de définitions

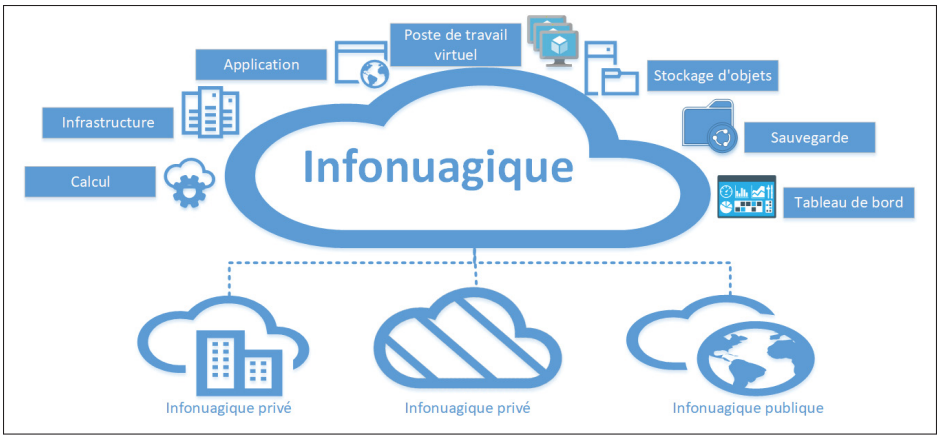


Figure 1.1 Les concepts de l'infonuagique

afin de les étudier et d'extraire un ensemble minimal de caractéristiques essentielles (Vaquero, Rodero-Merino, Caceres & Lindner, 2008). Dans ce contexte, il faut déterminer une définition claire et standard (OQLF, 2015; NIST & Technology, 2011).

La Figure 1.1 donne une représentation graphique de l'infonuagique. Selon l'Office québecois de la langue française, l'infonuagique est définie comme étant un (OQLF, 2015) : « *L'infonuagique est un modèle informatique qui, par l'entremise de serveurs distants interconnectés par Internet, permet un accès réseau, à la demande, à un bassin partagé de ressources informatiques configurables, externalisées et non localisables, qui sont proposées sous forme de services, évolutifs, adaptables dynamiquement et facturés à l'utilisation.* ».

D'après, la National Institute of Standards and Technology (NIST & Technology, 2011; Mell, Grance et al., 2011) : « *L'infonuagique est un modèle permettant un accès réseau omniprésent, convenable et à la demande à des ressources informatiques partagées et configurables (par exemple, réseaux, serveurs, stockage, applications et services) qui peuvent être rapidement provisionnés et libérés avec un effort de gestion ou interaction minimal avec le fournisseur de services.* ».

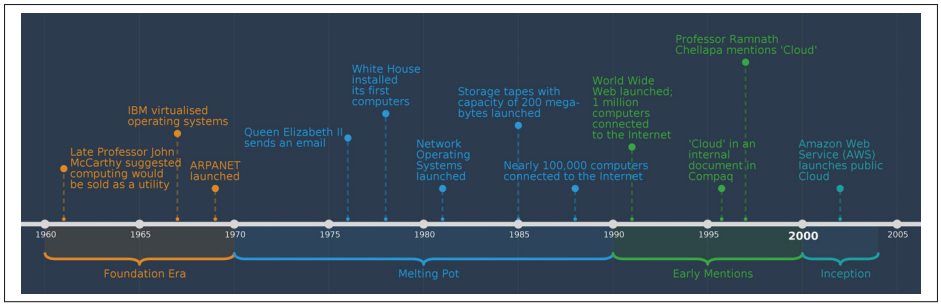


Figure 1.2 Chronologie de l'infonuagique entre 1960 et 2005
Tiré de BCS (2019)

1.2.2 Chronologie de l'infonuagique

L'arrivée de l'infonuagique a été le résultat d'une évolution progressive à plusieurs étapes (voir la Figure 1.2). Dans les années 1950, l'informatique a connu le développement des applications utilisant des systèmes centraux partagés (appelé : mainframes). Lors de cette époque, plus précisément en 1961, le Professeur John McCarthy a été le premier à prédire publiquement (dans un discours prononcé pour célébrer le centenaire du MIT) que les ressources informatiques pourraient être offertes comme un service public tel que l'eau ou l'électricité : "*Computing may someday be organized as a public utility just as the telephone system is a public utility! Each subscriber needs to pay only for the capacity he actually uses, but he has access to all programming languages characteristic of a very large system. Certain subscribers might offer service to other subscribers. The computer utility could become the basis of a new and important industry*."

L'idée de développer un réseau informatique mondial a été introduite en 1962 par Joseph Carl Robnett Licklider du Massachusetts Institute of Technology (MIT). Il a géré un programme en collaboration avec la Defense Advanced Research Projects Agency (DARPA) qui a abouti par la suite au développement du premier réseau de commutation par paquets aux États-Unis appelé ARPANET (Advanced Research Projects Agency Network). La première démonstration de l'ARPANET date d'octobre 1972.

En parallèle, dans les années 1970, IBM a développé un système d'exploitation appelé VM qui permettait aux administrateurs systèmes de l'IBM System/370 d'avoir plusieurs Machines Virtuelles (VM) sur un seul nœud physique. Le système d'exploitation VM a permis l'exécution simultané de plusieurs systèmes d'exploitation dans des environnements isolés.

La première définition connue du terme «Cloud Computing» est celle du professeur Ramnath Chellappa à Dallas en 1997 (Chellappa, 1997) : "*A computing paradigm where the boundaries of computing will be determined by economic rationale rather than technical limits alone*.".

En 1999, l'éditeur de logiciels 'Salesforce' était le pionnier à proposer un système de gestion de la relation client (CRM) à travers l'internet.

Au début des années 2000, a connu le développement de plusieurs hyperviseurs publics qui permettront l'exécution simultanée de plusieurs systèmes d'exploitation sur une seule machine. Après VMWare, Citrix et Microsoft , Red Hat a lancé en 2008 le premier hyperviseur public Open Source (Caizergues, 2008).

En 2006, Amazon a élargi ses services cloud. Au début, il propose un service appelé 'Cloud Elastic Compute (EC2)', qui offre aux utilisateurs l'accès à des ordinateurs et d'exécuter leurs propres applications via l'internet. En plus, il propose un autre service appelé 'Simple Storage Service (S3)'. Cela a introduit le modèle de paiement à l'utilisation appelé 'pay-as-you-go'.

En 2014, les dépenses des entreprises mondiales pour les infrastructures et les services infonuagiques ont atteint environ 34,1 milliards de dollars américains, avec une hausse de 13% par rapport au montant dépensé en 2013 (Statista, 2020). Tandis qu'en 2020, Gartner prévoit une croissance des revenus mondiaux des services infonuagiques publiques de 6,3% (Gartner, 2020a).

1.2.3 Les différents services infonuagiques

Généralement, il existe plusieurs formes pour externaliser les services et infrastructures informatiques afin de les rendre accessibles à distance via les réseaux Internet. La différence entre ces types d'infonuage est fonction du degré d'externalisation souhaité des ressources.

Selon la National Institute of Standards and Technology (Mell *et al.*, 2011), il existe principalement trois modèles de services infonuagique : IaaS (*Infrastructure as a Service*), PaaS (*Platform as a Service*) et SaaS (*Software as a Service*). La Figure 1.3 illustre une représentation graphique de ses trois principaux modèles.

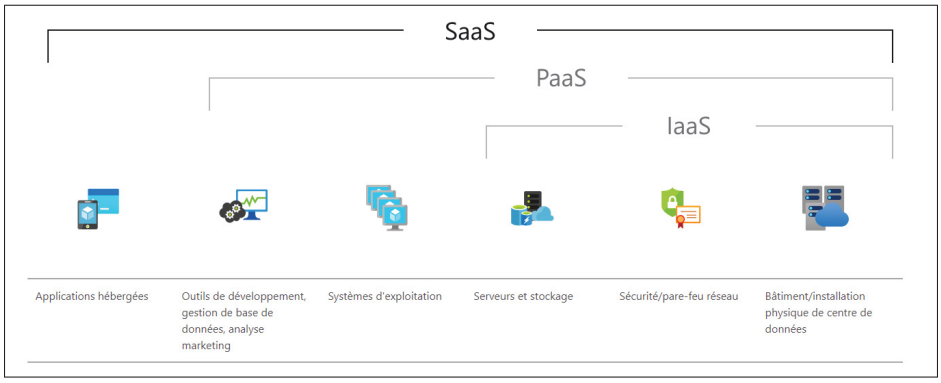


Figure 1.3 Les différents services infonuagiques
Tiré de Microsoft (2021)

1.2.3.1 IaaS – Infrastructure as a Service

IaaS (*Infrastructure as a Service*) appelée aussi infrastructure en tant que service. Cette infrastructure fournit la capacité de traitement, stockage, réseaux ainsi que d'autres ressources informatiques. Elle permet aux consommateurs de déployer et d'exécuter des systèmes d'exploitation ou des applications.

L'infonuage IaaS signifie l'externalisation de l'infrastructure matérielle du service informatique (réseaux, stockage et serveurs) chez un fournisseur externe comme le démontre la Figure 1.4.

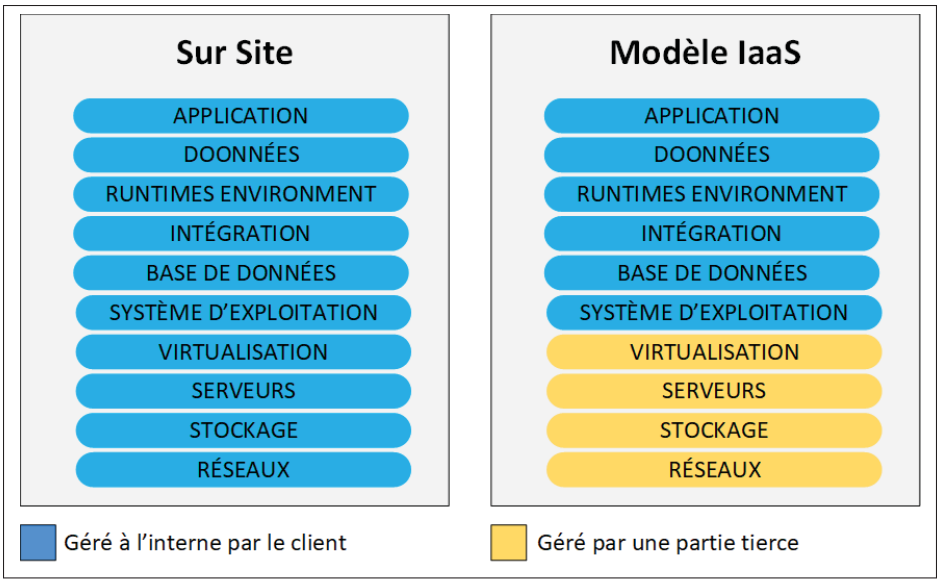


Figure 1.4 Représentation des responsabilités du modèle IaaS

Les fournisseurs de ce type d'infonuage interdisent ou limitent aux consommateurs la gestion et le contrôle de l'infrastructure puisque ce sont les fournisseurs qui s'occupent de cette tâche, tandis que leurs clients se concentrent sur l'installation, la configuration et la gestion d'intergiciel de l'infrastructure ainsi que les logiciels applicatifs (par exemple des serveurs d'applications, web ou de bases de données).

Les services d'infrastructure fournis par les fournisseurs à travers l'infonuage permettent à leurs clients d'approvisionner assez facilement un grand nombre d'instances de calcul (Ekanayake & Fox, 2010). IaaS sert de couche de base pour les autres types de services infonuagiques (Bhardwaj, Jain & Jain, 2010).

L'infonuage IaaS permet aux utilisateurs finaux d'évoluer et de réduire les ressources en fonction des besoins, réduisant ainsi le besoin de dépenses d'investissement initiales élevées et inutiles. Contrairement aux autres modèles, l'IaaS offre le contrôle le plus bas des ressources dans l'infonuage. La popularité de ce modèle a augmenté au début des années 2010. Cependant, avec la venue des nouvelles technologies, telles que les conteneurs, le sans-serveur (*serverless*) et la

montée en puissance des microservices, l'infonuage IaaS reste fondamental et demeure très en demande (IBM Cloud Education, 2019).

Selon Gartner, le marché mondial de l'infonuagique IaaS a atteint 44,5 milliards de dollars américains, avec une progression de 37,3% par rapport aux 32,4 milliards de dollars américains enregistrés en 2018 (Gros, 2020).

L'infonuage IaaS est composé essentiellement de plusieurs éléments (Rittinghouse & Ransome, 2016; Reese, 2009) :

- Serveurs (physiques et virtuels) ;
- Systèmes de stockage en utilisant un réseau de stockage (SAN), ou un stockage connecté au réseau (NAS) ;
- Réseau de communication (incluant des routeurs, commutateurs, pare-feu, équilibreur de charge, etc) ;
- Connectivité internet haut débit ;
- Environnement de virtualisation de plate-forme ;
- Service de nom de domaine (DNS), protocole de configuration dynamique d'hôte (DHCP) et autres services de gestion et d'assistance ;
- Sécurité en utilisant un pare-feu physique ou/et virtuelle et un système de détection et de prévention des intrusions ;
- Équilibreur de charge matérielle ;
- Alimentation, refroidissement et système de reprise après désastre.

1.2.3.2 PaaS – Platform as a Service

L'infonuage PaaS est un environnement de développement et de déploiement dans l'infonuage afin de permettre de fournir des services applicatifs. Les clients font l'acquisition des ressources dont ils ont besoin auprès d'un fournisseur de services infonuagiques avec un paiement à l'utilisation à travers une connexion internet. Comparativement à l'infonuage IaaS, l'utilisateur ne contrôle

pas l'intergiciel de l'infrastructure, les bibliothèques et environnements de développement ou d'exécution installés et les plateformes qui permettent l'hébergement et le déploiement des applications .

L'infonuage PaaS signifie l'externalisation de la gestion de la plateforme d'exécution des applications, des serveurs (physiques et virtuels), des logiciels de base (par exemple les systèmes d'exploitation et les serveurs de bases de données) et de l'infrastructure (par exemple les équipements du réseau, de stockage, de sauvegarde). Ainsi, le client gère seulement ses applications et ses données. La Figure 1.5 schématise les ressources gérées par le client de l'infonuagique PaaS par rapport à ceux gérés par le fournisseur de l'infonuage.

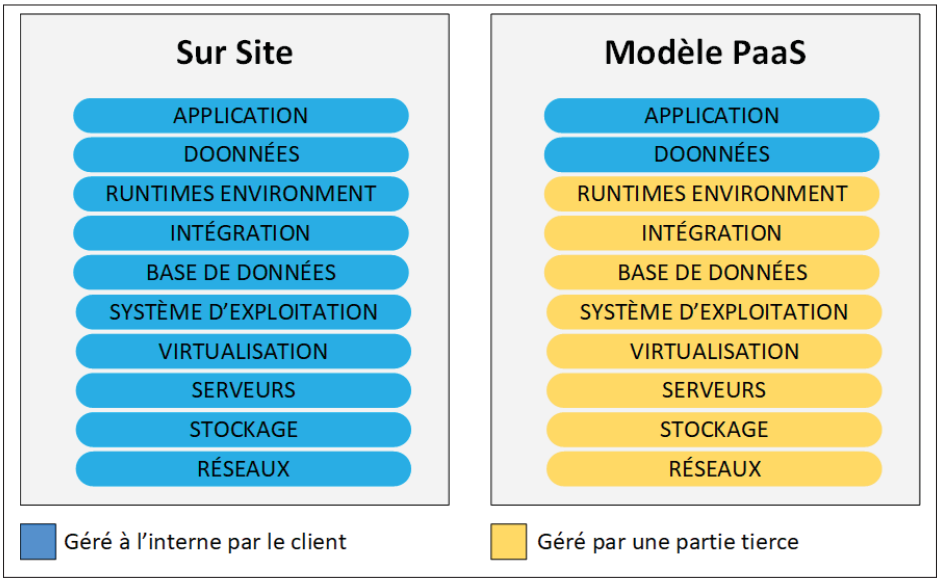


Figure 1.5 Représentation des responsabilités du modèle PaaS

Un exemple connu de ce type de déploiement est le 'Google Apps Engine' (Gupta, Goswami, Chaudhary & Bansal, 2020; Ifrah, 2021; Reese, 2009). Pour faciliter, le développement, différents outils sont mis à la disposition des clients, tels que les langages de programmation et les interfaces de programmation d'application (API).

1.2.3.3 SaaS – Software as a Service

L'infonuage SaaS, ou logiciel en tant que Service, est un modèle de distribution de logiciel standard hébergé sur les serveurs d'un fournisseur infonuagique. Les applications sont mises à la disposition des clients par des fournisseurs et accessible via Internet.

Les applications sont accessibles à partir d'une interface client comme les navigateurs Web. Les clients du modèle SaaS ne contrôlent ni l'infrastructure utilisée, ni les plateformes qui permettent l'hébergement et le déploiement des applications, à l'exception du paramétrage et de la configuration d'applications spécifiques au client. La Figure 1.6 présente la répartition des responsabilités entre le client et le fournisseur de l'infonuage SaaS.

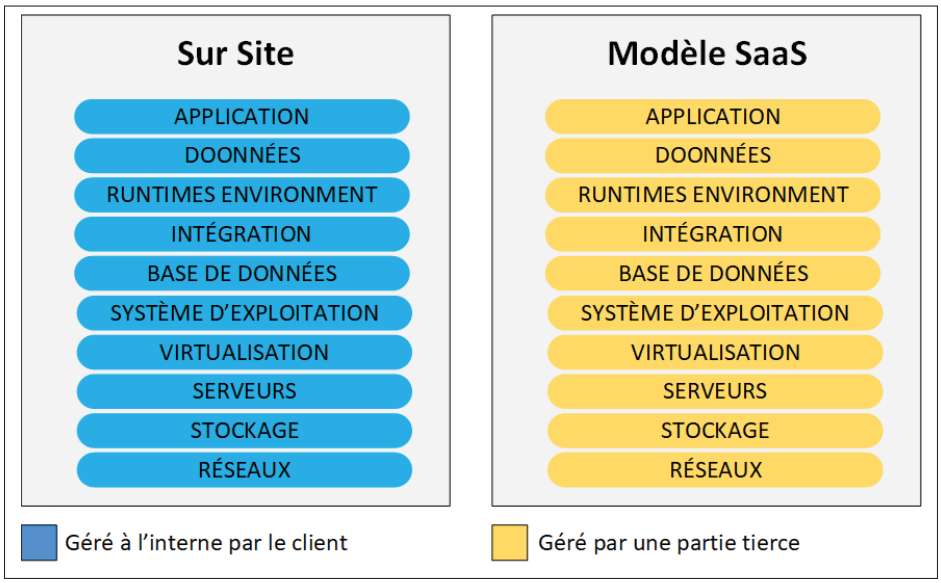


Figure 1.6 Représentation des responsabilités du modèle SaaS

Bien que le client perd certains niveaux de contrôle, le modèle SaaS transfère le fardeau de l'acquisition et du maintien d'une application d'entreprise au fournisseur du cloud. Il permet aux utilisateurs de tirer profit des fonctionnalités offertes par les solutions applicatives sans s'occuper eux-mêmes du déploiement ou de la gestion des ressources matériels ou logiciels associés.

Au lieu de se procurer des licences, d'installer et de maintenir des solutions applicatives standards sur les ordinateurs et les serveurs des clients, le modèle SaaS permet aux utilisateurs de bénéficier les logiciels des fournisseurs via l'internet selon le modèle de paiement à l'utilisation "Pay-as-you-go". Ce modèle de facturation le rend plus accessible aux organisations qui ne disposent pas des ressources nécessaires pour acheter, déployer et gérer l'infrastructure et les solutions applicatives nécessaires.

L'infonuage SaaS facilite la mobilité du personnel, puisque les utilisateurs peuvent accéder aux applications et aux données depuis n'importe quel endroit connecté à l'internet.

Google Docs est un exemple d'application de traitement de texte proposée sous forme d'un service infonuagique SaaS. L'utilisateur accède et utilise la solution via un navigateur Web, qui lui permet de l'utiliser via internet (Yadegaridehkordi, Nilashi, Shuib & Samad, 2020; Yadegaridehkordi, Nilashi, Shuib, Asadi & Ibrahim, 2019). À la différence des autres services infonuagiques (PaaS et IaaS), avec le SaaS, l'utilisateur ne peut modifier ni l'application, ni le matériel sur lequel l'application s'exécute, ni la configuration du réseau.

1.2.3.4 Autres types de services infonuagiques

Trois modèles les plus importants de services infonuagique sont le PaaS, le SaaS et le IaaS. Cependant, d'autres modèles dérivés ont apparu récemment comme (Martin, 2020; Chang, Li & Ranganathan, 2020; Kamongi, 2019; Wada, 2018; Éditorial Geekflare, 2020) :

• **DaaS - Data as a Service :** Data as a service (DaaS), est un modèle qui offre l'accès à un dépôt de données via une interface. Le service proposé par les fournisseurs dans ce cas est les données, qui peuvent être statistiques sur un marché spécifique et qui vont permettre de prendre la meilleure stratégie commerciale. Ce modèle utilise l'infonuage pour offrir le stockage de données, l'intégration de données et des services d'analyse de données (Tibco, 2021b; Avi, 2020).

• **SECaaS - Security as a Service :** L'infonuage SECaaS vise à offrir des solutions de sécurité informatique par des fournisseurs externes via le Web. Ce type de déploiement permet aux organisations de se protéger contre les risques de sécurité en déviant les flux de données entrants vers un fournisseur qui les filtre avant de le transmettre à nouveau vers l'organisation, comme le démontre la Figure 1.7. Ce modèle propose plusieurs services, comme l'anti-malware, le pare-feu, la détection d'intrusion, l'authentification, les tests d'intrusion, l'anti-spam et la gestion des identités.

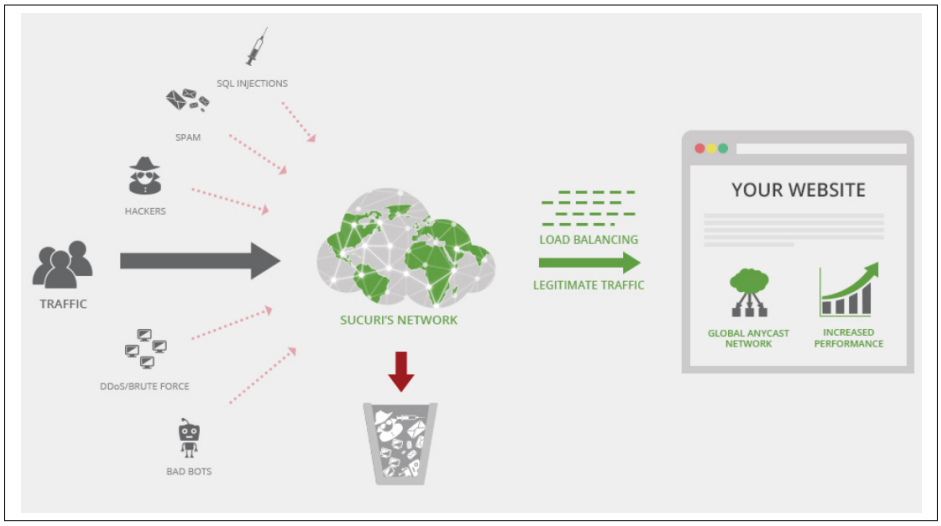


Figure 1.7 SECaaS - Security as a Service
Tiré de Avi (2020)

• **APIaaS - API as a service :** L'infonuage APIaaS permet la création et l'hébergement d'interface de programmation applicative (API) et aussi la connexion à des APIs tierces comme Google map, PayPal API et Bloomberg API. Ces APIs permettent ainsi aux utilisateurs de communiquer avec les fonctionnalités offertes par des applications dorsales (*backend*). La Figure 1.7 illustre les composantes et le fonctionnement de l'infonuage APIaaS.

• **DaaS - Desktop as a Service :** Ce type de service permet d'offrir des bureaux virtuels hébergés sur l'infonuage et accessible partout à travers l'internet. Il permet d'offrir une plus grande mobilité du personnel.

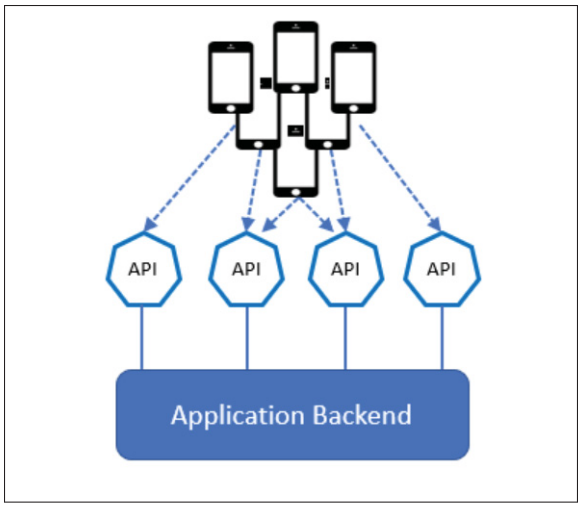


Figure 1.8 APIaaS - API as a Service
Tiré de Avi (2020)

• **DBaaS - Database as a Service :** Dans le cadre du modèle DBaaS, le fournisseur gère l'infrastructure et la base de données et délègue la gestion du contenu et des opérations au client. Il fournit des fonctionnalités de base de données en tant que service aux clients internes ou externes.

• **BaaS - Backend as a service :** Backend-as-a-Service (BaaS) est un modèle de service infonuagique qui prend en charge les services dorsaux (Backend) d'une application, tandis que les développeurs peuvent se concentrer uniquement sur le codage et la maintenance de la partie frontal (frontend) de l'application. Les exemples de services dorsaux sont l'authentification des utilisateurs, la gestion de bases de données, la mise à jour à distance et les notifications, ainsi que le stockage et l'hébergement dans le cloud (Tibco, 2021a). La Figure 1.9 présente la répartition des responsabilités entre le client et le fournisseur de l'infonuage BaaS.

• **FaaS - Function as a Service :** L'infonuage FaaS permet aux clients de développer, d'exécuter et de gérer les fonctionnalités dans des conteneurs. Le FaaS utilise le principe de l'informatique sans serveur, en se basant sur une logique qui s'exécute dans des conteneurs entièrement gérés par le fournisseur de plateforme infonuagique (Redhat, 2021).

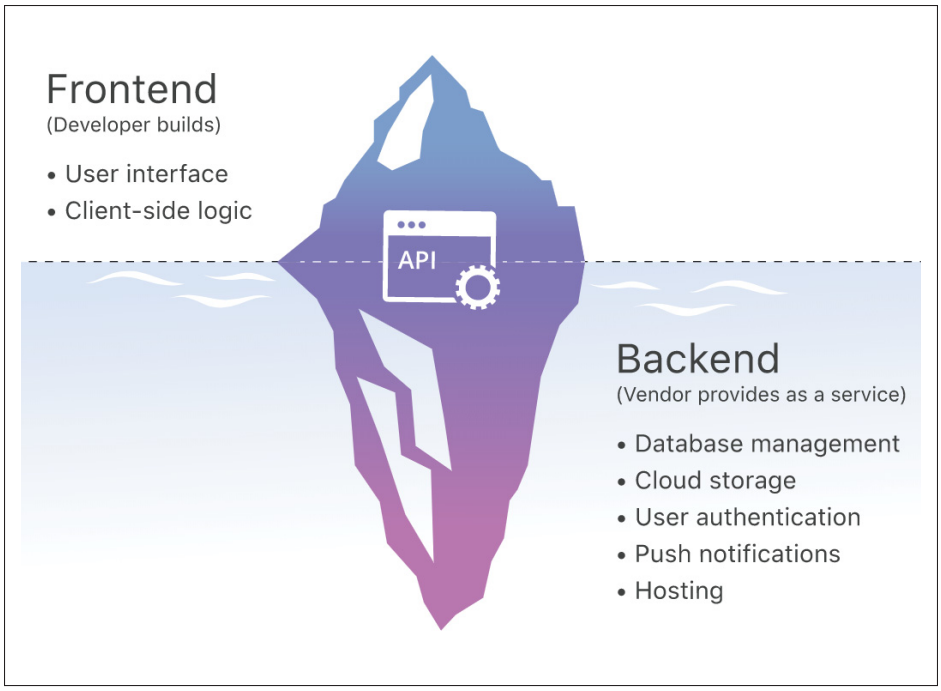


Figure 1.9 BaaS - Backend as a service
Tiré de Tibco (2021a)

• **AaaS - Analytics as a Service :** L'infonuage AaaS est un service d'analyse infonuagique. Ce type de modèle offre plusieurs services, comme l'analyse prédictive, l'analyse de données et l'analyse commerciale afin de mieux interpréter la tendance de données analysées et de prendre les meilleures décisions.

1.2.4 Les modèles de déploiement de l'infonuagique

Il existe différents modèles de déploiement de l'infonuagique selon le groupe cible visé, l'organisation et la provenance des services. On dénombre quatre types d'infonuage à savoir l'infonuage public, l'infonuage privé, l'infonuage hybride et l'infonuage communautaire (Mell *et al.*, 2011).

1.2.4.1 L'infonuage public

L'infonuage public est mis à la disposition du grand public ou d'un grand groupe industriel par un fournisseur de services infonuagiques (Mell *et al.*, 2011). Ce type d'infonuage est offert au grand public comme aux grandes organisations et géré par un fournisseur externe qui offre ce service (Mell & Grance, 2009). Les informations créées et soumises par les utilisateurs sont généralement stockées sur les serveurs du fournisseur externe (Hamrén, 2012). Il est hébergé sur l'internet et accessible par le grand publique et les organisations à travers le réseau d'un fournisseur de services Internet. L'infonuage public permet ainsi de réduire à la fois les dépenses d'exploitation (OPEX) et les dépenses en investissement (CAPEX).

Une des caractéristiques de l'infonuagique publique est que les utilisateurs ne connaissent nécessairement pas où sont stockées réellement leurs données ni comment elles sont sauvegardées et encore moins si d'autres utilisateurs non autorisés peuvent y accéder.

Selon le rapport Gartner, les trois leaders mondiaux de l'infonuagique publique pour l'année 2020 sont Amazon, Google et Microsoft (Gartner, 2020b). La Figure 1.10) montre le classement des fournisseurs de l'infonuagique public pour l'année 2020.

Selon Gartner (Gartner, 2020c), le marché mondial des services de l'infonuagique public devrait croître de 6,3% en 2020 pour atteindre 257,9 milliards de dollars américains, comparativement à 242,7 milliards de dollars américains en 2019.

Ce type d'infonuagique offre plusieurs avantages :

- Agilité : répondre rapidement aux demandes imprévisibles ;
- Flexibilité des coûts : en fonction des niveaux de services demandés (*Service Level Agreement* - SLA) ;
- Grande évolutivité à la demande : pas besoin de déployer de nouveaux serveurs ou réseaux ;
- Rentabilité : faibles coûts (être dispensé des frais d'achat, de gestion et de la maintenance des matériels informatiques et des logiciels ;
- Fiabilité : support technique 24/7.

Figure 1.10 Classement des fournisseurs de l'infonuagique public pour l'année 2020
Tiré de Gartner (2020b)

Cependant, les principaux inconvénients sont principalement les risques de manque de garantie sur la sécurité et la confidentialité des données ;

1.2.4.2 L'infonuage privé

L'infonuage privé est dédié uniquement à une seule et même organisation. Il peut être géré par l'organisation elle-même ou géré par un tiers et peut être hébergé à l'interne (appelé : infonuage privé interne) ou à l'externe (appelé : infonuage privé externe) (Mell *et al.*, 2011). L'infonuage est accessible uniquement par les membres de l'organisation et, dans certains cas, par des collaborateurs autorisés. L'objectif est de partager les ressources informationnelles au sein de la même organisation. Par exemple, une entreprise qui possède plusieurs sites à travers le globe et qui souhaite mettre des données et des outils communs à la disposition de ses différents sites.

L'inconvénient majeur de l'infonuage privé est son coût total de possession (*Total cost of ownership* - TCO) très élevé. Plusieurs études démontrent que le coût d'achat d'équipements, de logiciels et de personnel entraîne souvent des coûts plus élevés pour une organisation disposant de son propre infonuage privé comparativement à l'infonuage public (Goyal, 2014).

L'infonuage privé donne à l'organisation un plus grand contrôle sur l'infrastructure et les ressources de calcul (Jansen, Grance et al., 2011). Ce type d'infonuage répond plus aux enjeux de la sécurité et de la confidentialité des données comparativement à l'infonuage public.

D'un côté, les ressources (par exemple : humain, financier, informationnel et matériel) et les données représentent sans doute les éléments les plus importantes de toute organisation. Ainsi, confier ces ressources à des parties tierces place les organisations dans une situation fragile et vulnérable (Veronica, 2012; Ghanbari, Simmons, Litoiu & Iszlai, 2012).

1.2.4.3 L'infonuage hybride

L'infonuage hybride est plus complexe que les autres modèles de déploiement, puisqu'il est le résultat de la composition de deux ou plusieurs types d'infonuage (privés, communautaires ou publics). Chaque membre reste une entité unique, mais il est lié aux autres à travers une technologie standard ou propriétaire qui permet la portabilité des applications et des données entre eux (Jansen *et al.*, 2011).

Durant ces dernières années, plusieurs petites organisations sont devenues dépendantes d'un environnement infonuagique hybride qu'elles ont conçu au hasard afin de répondre à leurs besoins commerciaux (Jansen *et al.*, 2011). Compte tenu de la popularité croissante et de l'adoption généralisée de ce type de déploiement, le marché de l'infonuagique hybride devrait augmenter (Kelly, Furey & Curran, 2021). Il n'y a pas deux infonuages hybrides identiques et ce type d'infonuage est peu normalisé, ce qui crée plus de défis lors du déploiement (Jansen *et al.*, 2011).

Selon un rapport de recherche (MarketsandMarkets, 2018), le marché de l'infonuagique hybride devrait passer de 44,6 milliards de dollars américains en 2018 à 97,6 milliards de dollars américains d'ici 2023, à un taux de croissance annuel composé (*Compound Annual Growth Rate* - TCAC) de 17,0%. L'augmentation de la demande est motivée par plusieurs facteurs, tels que la rentabilité, l'évolutivité, l'agilité et la sécurité.

Avec ce déploiement, une organisation utilise son propre infrastructure informatique pour répondre à ses besoins normaux, mais elle fait appel à l'infonuage public pour répondre aux besoins avec des exigences élevées. Il soulève de nombreux inconvénients tels que l'interopérabilité et la normalisation. Zhang et al. proposent d'effectuer les activités critiques au sein de l'infonuage privé de l'organisation et de déléguer les activités non critiques à l'infonuage public (Zhang, Cheng & Boutaba, 2010). Par exemple, certaines organisations hébergent leurs données de ressources humaines et de gestion de la relation client dans un infonuage public comme 'Saleforces.com' mais gardent leurs données confidentielles dans leur propre infonuage privé (Sarna, 2010).

L'approche hybride permet à une organisation de profiter de l'évolutivité et de la rentabilité qu'offre l'infonuage public sans exposer les applications et les données critiques aux risques de vulnérabilité (Goyal, 2014).

Ce type d'infonuagique offre une combinaison des avantages de l'infonuage public et privé :

- Il améliore l'allocation des ressources selon le besoin ;
- Il permet d'optimiser les coûts, par exemple en fonction l'importance, niveaux de services exigés, la confidentialité et le cycle de vie de l'infrastructure logiciel ou matériel ;
- Il offre le contrôle complet à travers l'infonuage privé et l'évolutivité de l'infonuage public.

Cependant, les inconvénients majeurs sont les suivants :

- Le risque de sécurité lors de l'extension de l'approche de résolution d'identité d'entreprise vers l'infonuage public représente ;
- Le risque de confidentialité et d'intégrité par rapport à la transition de données entre les deux environnements ;

- Le risque de non-conformité entre la politique de sécurité de l'infonuage public versus privé.

1.2.4.4 L'infonuage communautaire

L'infonuage communautaire est un environnement infonuage partagé par une communauté ou plusieurs communautés qui ont un objectif commun. Il est similaire à l'infonuage privé, mais l'infrastructure et les ressources de calcul sont réservé à deux organisations ou plus au sein d'une communauté en matière de confidentialité, de sécurité et de réglementation (Jansen *et al.*, 2011).

Selon Gartner (Gartner, 2021), infonuage communautaire est un environnement de service infonuagique partagé qui cible un ensemble limité d'organisations ou d'employés. Le principe d'organisation de la communauté varie, mais les membres de la communauté partagent généralement des exigences similaires en matière de sécurité, de confidentialité, de performances et de conformité. Les membres de la communauté peuvent mettre un mécanisme qui est souvent géré par eux-mêmes pour examiner les demandes des nouveaux membres qui souhaitent intégrer la communauté.

L'infonuage communautaire vise à combiner et unifier l'approvisionnement en ressources distribuées de plusieurs entités afin de fournir plus de performance aux organisations de la communauté comparativement à un fonctionnement en « silo ». Le but est mieux exploiter les ressources sous-utilisées pour former un infonuage communautaire (Marinos & Briscoe, 2009; Briscoe & Marinos, 2009).

Ce type d'infonuagique offre une combinaison des avantages de l'infonuage public et privé tels que (Marinos & Briscoe, 2009) :

- Il est moins coûteux que l'infonuage privé puisqu'il y a un partage des coûts entre les organisations de la communauté ;
- Les éléments déployés dans l'infonuage communautaire peuvent faciliter la collaboration entre la même communauté, par exemple mieux servir les consommateurs, rendre plus efficace la chaîne d'approvisionnement et meilleur suivi des opérations.

Ce type d'infonuage présente aussi certains inconvénients :

- Plus coûteux que l'infonuage public ;
- Ressources limitées : la bande passante et le stockage de données sont partagés entre tous les membres de la communauté.

1.2.5 Les avantages de l'infonuagique

D'une façon générale, les principaux avantages de l'infonuagique peuvent être résumés comme suit :

1. **Réduire les dépenses OPEX et CAPEX globaux :** L'infonuage permet aux organisations de réduire les dépenses d'exploitation (OPEX) liées aux charges courantes pour exploiter par exemple le réseau ou les serveurs et aussi les dépenses d'investissement (CAPEX) d'une organisation pour l'achat d'infrastructure.
2. **Évolutif, fiable et élastique :** Les services en nuage sont élastiques et évolutifs puisqu'ils offrent la possibilité de redimensionner ou de réduire les ressources nécessaires selon le besoin et même au cours de l'utilisation. Cela offrira une meilleure efficacité et permettra d'empêcher l'inutilisation des ressources. L'utilisateur des services infonuagiques pourra facilement et à tout moment ajuster par exemple : la bande passante requise, la vitesse de traitement ou le nombre de licences. L'infonuagique est considérée comme étant plus fiable que l'infrastructure informatique interne. Avec l'utilisation de plusieurs sites redondants, la fiabilité est assurée à travers le nuage. La haute disponibilité fait de l'infonuage une solution parfaite pour le recouvrement en cas de désastre ou pour exécuter des tâches critiques (Rashid & Chaturvedi, 2019).
3. **Rapidité et efficacité d'implémentation :** L'infonuage permet aux organisations de disposer de plus de ressources rapidement afin de répondre à une forte demande. La rapidité et l'efficacité sont considérées comme un important avantage de l'infonuagique afin de favoriser la transformation numérique. En plus, les petites organisations pourront bénéficier de l'utilisation des applications d'analyses commerciales avec des calculs intenses, chose qui était strictement résrvé dans le passé que pour les plus grandes organisations.

4. **Facturation basée sur la consommation :** Avec l'utilisation des services infonuagiques, les organisation n'ont besoin de dépenser que ce qu'elles utilisent et quand elles ont besoin d'augmenter leur capacité, selon un modéle de facturation à l'utilisation (Mydyti, Ajdari & Zenuni, 2020).
5. **Partage, Mobilité et Internet :** L'infonuage offre un gain en productivité à travers le partage, la collaboration et le stockage de fichiers d'une manière sécurisé, ainsi qu'un l'accès à ces derniers à tout moment et en tout lieu. Ces fonctionnalités seront accessibles en ligne via un accès internet ou un accès VPN.
6. **Sécurité de données :** La sécurité est l'un des facteurs les plus déterminants pour l'adoption de l'infonuage lors de toute transformation numérique. L'aspect important pour les organisations qui souhaitent franchir le pas vers l'infonuage, est la capacité des fournisseurs de services infonuagiques à assurer la sécurité des données, des communications ainsi que les applications utilisées.

1.3 La virtualisation

1.3.1 Définition

La virtualisation est un concept qui permet l'abstraction du matériel physique sous forme de logiciel afin de fournir plusieurs environnements ou ressources utiles indépendamment des spécificités des plate-formes matérielles. L'objectif est de maximiser l'utilisation du matériel physique en répartissant les ressources informatiques entre les différents services informatiques offerts par la même infrastructure physique.

Comme le montre la Figure 1.11, les deux principaux composants de la virtualisation d'un serveur sont la Couche matérielle et l'hyperviseur. La couche matérielle est le matériel physique utilisé pour fournir plusieurs environnements ou ressources. L'hyperviseur est la solution logicielle de virtualisation installée sur une machine hôte pour permettre l'exécution de plusieurs machines virtuelles sur la même machine physique. Les hyperviseurs peuvent fonctionner sur un système

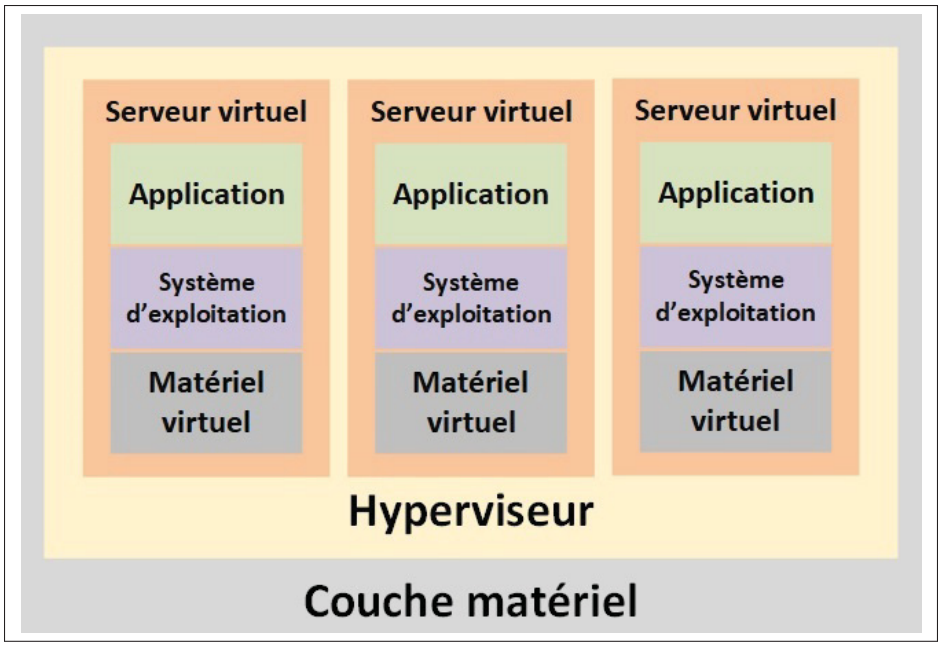


Figure 1.11 Virtualisation des serveurs

d'exploitation ou bien directement installé sur une couche matérielle, cette deuxième option est adoptée à la majorité par les organisations.

Il existe deux grandes familles d'hyperviseurs :

- Hyperviseur de type 1 : comme le montre la Figure 1.12, c'est une solution logiciel qui s'installe directement sur le serveur. Ces solutions prennent en charge le contrôle du serveur physique, allouent les ressources et gèrent les systèmes d'exploitation invités des machines virtuelles qu'elles hébergent. Ce type à l'avantage d'exploiter toutes les ressources de l'équipement physique. Cependant, il n'est pas possible d'exécuter qu'un seul hyperviseur par serveur physique.
- Hyperviseur de type 2 : cette solution logicielle s'exécute sur un système d'exploitation d'un serveur physique comme l'illustre la Figure 1.13. Ce type a l'avantage de pouvoir héberger plusieurs hyperviseurs simultanément. Cependant, les ressources de la couche physique sont partagées entre le système d'exploitation de la machine hôte et les hyperviseurs qu'ils contiennent.

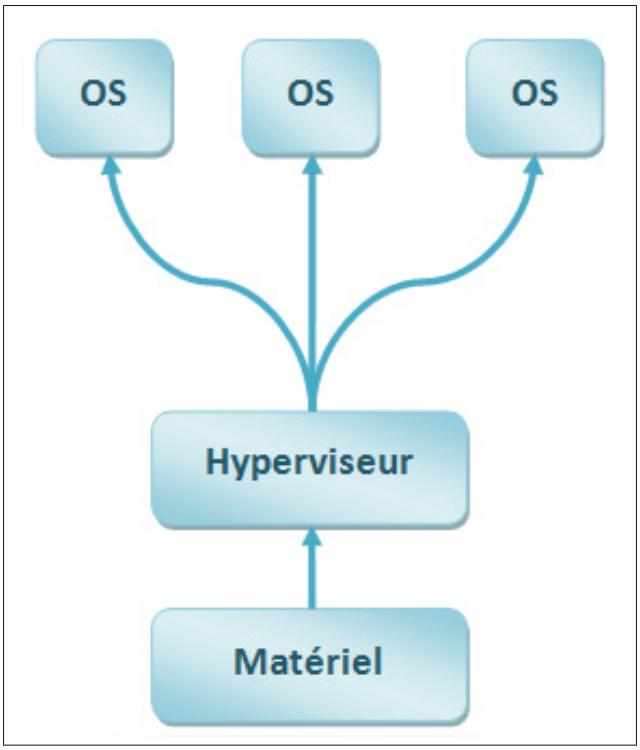


Figure 1.12 Hyperviseur de type 1
Tiré de It-connect (2021)

1.3.2 Types de virtualisation

Il existe principalement cinq types de virtualisation : la virtualisation des serveurs (émuler un système d'exploitation), du stockage, des applications, des postes de travail et du réseau.

1. **Virtualisation des serveurs :**La virtualisation des serveurs permet d'émuler plusieurs systèmes d'exploitation sur le même serveur. Ainsi, elle permettra d'utiliser efficacement les ressources du serveur physique.
2. **Virtualisation du stockage :** Ce type de virtualisation permet l'abstraction de plusieurs disques de stockage physiques sous forme d'un seul stockage logique pour toute l'organisation.
3. **Virtualisation du réseau :** permets de créer un réseau défini par logiciel (SDN) complet qui représente une abstraction logique des ressources réseau physique.

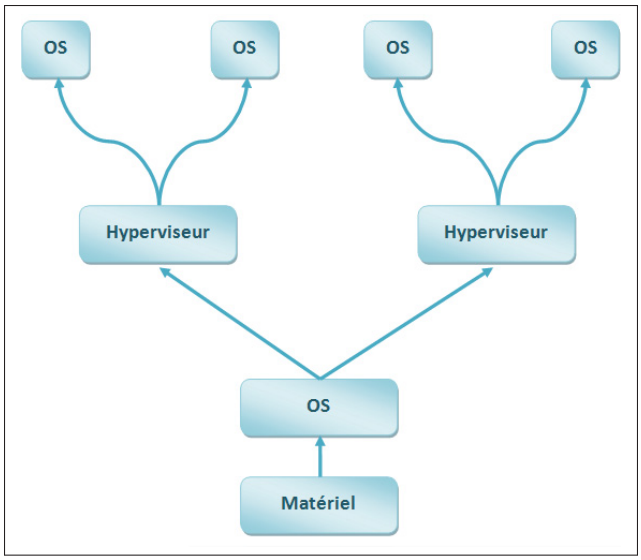


Figure 1.13 Hyperviseur de type 2
Tiré de It-connect (2021)

4. **Virtualisation du poste de travail :** offre la possibilité de déployer plusieurs postes de travail sur des serveurs distants, afin de permettre l'accès au bureau de travail de n'importe quel équipement ou emplacement.
5. **Virtualisation des applications :**permet d'utiliser l'application peu importe l'emplacement de l'utilisateur et élimine le besoin d'installer l'application. La suite Office 365 est un exemple concret de ce type de virtualisation d'applications, puisqu'elle offre l'accès à toutes les applications Office partout et sans installation.

1.4 La réseautique définie par logiciel

La réseautique définie par logiciel (*Software Defined Networking* - SDN) a été récemment proposée pour offrir plus de flexibilité aux opérateurs de réseau en leur permettant de configurer d'une façon rapide et dynamique les équipements réseau. Selon l'ONF (*Open Networking Foundation* - ONF) (ONF, 2020), la technologie SDN est définie comme étant une architecture émergente qui est en même temps dynamique, facile à gérer, rentable, évolutive et parfaitement adaptée à la nature dynamique des applications actuelles exigeant des bandes passantes élevées.

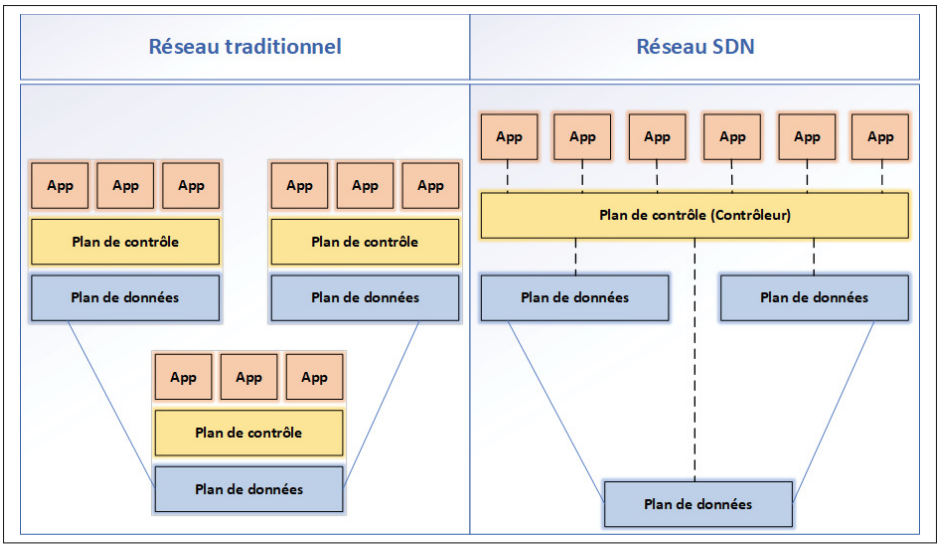


Figure 1.14 Comparaison entre un réseau traditionnel et un réseau défini par logiciel

En effet, dans une architecture réseau traditionnelle, il existe deux plans : 1) le plan de contrôle qui est responsable de la prise des décisions (par exemple, décisions de routage ou de gestion des priorités) et 2) le plan de données qui est responsable de la commutation des paquets. Comme le montre la Figure 1.14, dans les réseaux traditionnels, chaque équipement réseau implémente en même temps les deux plans. Le plan de contrôle de chaque équipement réseau est capable de construire les tables de routage en fonction des informations échangées avec les autres équipements. Ces tables sont utilisées par le plan de données de l'équipement pour commuter les paquets.

Contrairement aux réseaux traditionnels, dans un réseau SDN, le plan de contrôle est déplacé dans un nœud central appelé contrôleur (voir la Figure 1.14). Ce contrôleur bénéficie d'une vue globale du réseau et prend les décisions de routage pour tous les équipements réseau qui sont appelés les commutateurs SDN. Ainsi, ces commutateurs implémentent seulement le plan de données et reçoivent les décisions de routage de la part du contrôleur sous forme de règles de commutation. L'avantage principal de cette approche est la flexibilité et la programmabilité permettant au contrôleur de gérer plus facilement les équipements réseau et d'adapter dynamiquement la configuration du réseau avec les besoins des applications réseau.

Comme le montre la Figure 1.15, un réseau SDN est composé des trois couches suivantes :

1. Couche d'infrastructure (appelée aussi la couche de données) : cette couche est composée de commutateurs SDN et a pour rôle de transférer les paquets selon les règles de commutation établie par le contrôleur.
2. Couche de contrôle : cette couche s'occupe de la réception des requêtes de la couche d'application et de les transmettre à l'infrastructure réseau. Le contrôleur est responsable de la gestion des tables de routage du réseau. La communication entre le contrôleur et la couche d'infrastructure utilise une interface de programmation d'application appelée 'Southbound API' (par ex., le protocole OpenFlow). Le contrôleur est responsable de la gestion des tables de routage du réseau. La communication entre le contrôleur et la couche d'infrastructure utilise une interface de programmation d'application appelée Southbound API comme OpenFlow (OF) (ONF, 2020), ForCES (Halpern, Salim et al., 2010), NetConf (Enns, Bjorklund, Schoenwaelder & Bierman, 2011), IRS (Clarke, Salgueiro & Pignataro, 2016).
3. Couche application : cette couche est composée d'applications SDN qui définissent les stratégies de routage et de gestion du trafic dans le réseau. Ces applications SDN communiquent avec le contrôleur via une interface de programmation d'application appelée Northbound API. Les principales fonctions du contrôleur sont les suivantes. Il fournit à la couche d'application via les Northbound APIs la possibilité de programmer les fonctionnalités du réseau comme le routage, l'équilibrage de charge et la sécurité. Il collecte les informations sur la couche d'infrastructure et traduire les demandes de la couche d'application sous forme de règles à mettre en œuvre dans les nœuds physiques. De plus, il gère les demandes de la couche d'infrastructure et contrôle le niveau du plan de données (c.-à.-d., les commutateurs SDN) en utilisant une Southbound API comme OpenFlow.

Dans le cadre d'un réseau SDN utilisant OpenFlow, lorsqu'un commutateur reçoit un flux qui ne correspond pas aux règles dans sa table de flux, il envoie un message de type *PacketIn* au contrôleur (voir la Figure 1.16.a). Le contrôleur installe les règles de transfert nécessaires dans les commutateurs en envoyant un message de type *FlowMod* (voir la Figure 1.16.b). Le

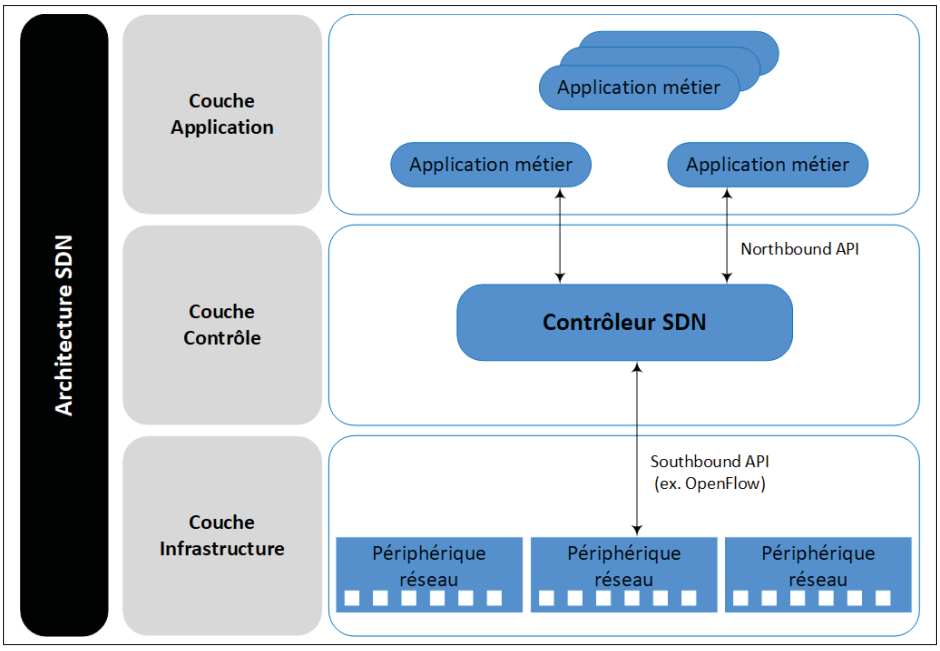


Figure 1.15 Les trois couches d'un réseau SDN

contrôleur peut spécifier un délai d'inactivité pour chaque règle de transfert, après quoi la règle est éliminée de la table de flux du commutateur. Quand une règle est éliminée, le commutateur envoie un message de type *FlowRemoved* au contrôleur. Ce message contient la durée du flux, ainsi que la taille du trafic correspondant à cette règle de commutation. En plus de ces messages, le contrôleur peut envoyer un message *FlowStatistics-Request* au commutateur pour interroger les statistiques d'un flux spécifique. Le commutateur envoie la durée et le nombre d'octets pour ce flux dans un message *FlowStatisticsReply* au contrôleur.

1.4.1 Avantages de la réseautique définie par logiciel

Cette nouvelle technologie offre plusieurs avantages :

- Programmabilité et flexibilité : la technologie SDN permet aux opérateurs de configurer et gérer dynamiquement le réseau via des interfaces de programmation. Ainsi, les changements de configuration des routeurs peuvent être effectués et appliqués d'une façon presque instantanée grâce au contrôleur et des *APIs southbound* offertes aux opérateurs réseau.

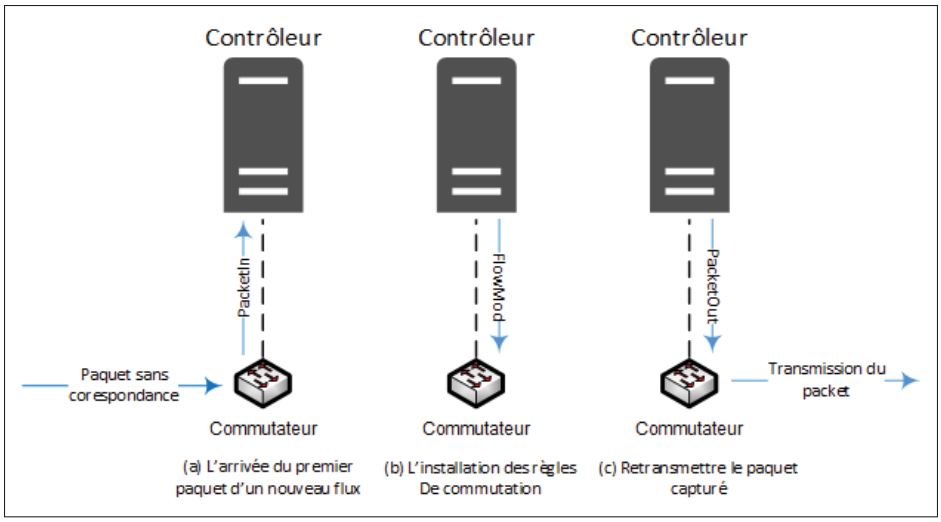


Figure 1.16 Les étapes d'installation d'une nouvelle règle OpenFlow

- Performance et mise à l'échelle (*scalability*) : grâce à la vue globale du réseau disponible au niveau de la couche de contrôle, il est facile d'optimiser les chemins suivis par les paquets et de les adapter en fonction du changement des besoins en termes de performance (par ex., délai, débit, utilisation). Ainsi, il est facile de reconfigurer les équipements réseau et de gérer dynamiquement les ressources utilisées afin de la gérer plus efficacement les éventuelles montées subites de la demande des utilisateurs.
- Intégration de service : cette architecture facilite l'intégration des nouveaux services réseautiques (par ex., l'équilibrage de charge et le contrôle d'admission) au niveau la couche application.
- Réduction des coûts : SDN permet de réduire les coûts d'utilisation d'infrastructure, en permettant aux utilisateurs de solliciter seulement les ressources réseaux dont ils ont besoin.
- Standard ouvert : grâce à cette nouvelle architecture, les équipements du réseau SDN peuvent être configurés avec les mêmes APIs quel que soit le fabricant de l'équipement.

Il est clair que cette technologie transforme complètement la façon avec laquelle les réseaux sont configurés et gérés. Elle permet, entre autres, une meilleure exploitation des ressources réseau en les ajustant dynamiquement avec les besoins grandissants des applications en termes

de performance et de flexibilité. Malgré ces avantages, plusieurs défis majeurs doivent être surmontés avant de pouvoir déployer cette technologie dans les réseaux à grande échelle.

1.5 Virtualisation des fonctions réseau

La virtualisation des fonctions de réseau (NFV) (Stallings, 2015) est un concept qui utilise la technologie de virtualisation pour approvisionner dynamiquement des fonctions réseau au lieu de les avoir dans des équipements dédiés (par ex., routeur, services d'annuaire, les inspecteurs de paquets (*Deep Packet Inspector* - DPI), les pare-feu) (Figure 1.17). Elle permet ainsi une grande flexibilité afin de fournir de nouveaux services réseau et de réduire le délai de la mise en service de ces fonctions réseau.

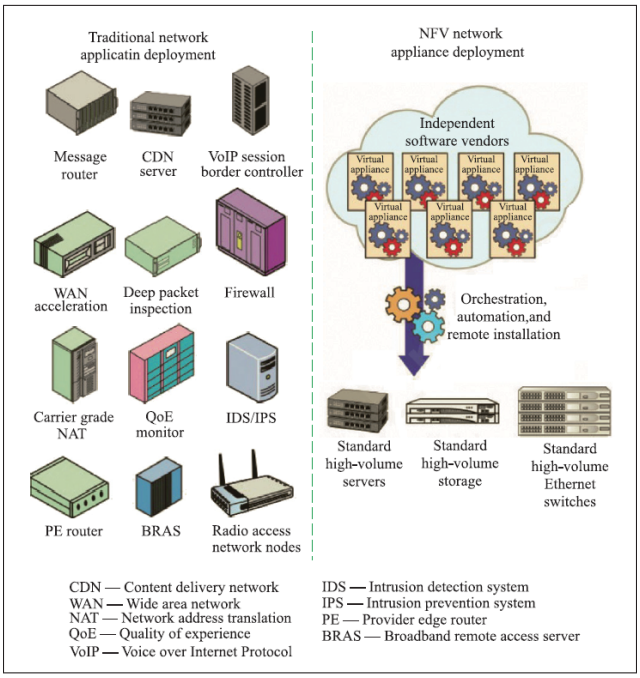


Figure 1.17 virtualisation des fonctions de réseau (NFV)
Tiré de Stallings (2015)

Le Tableau 1.1 présente quelques fonctions réseau qui sont susceptibles d'être virtualisées.

Le concept de la virtualisation des fonctions réseau offre plusieurs avantages tels que :

Tableau 1.1 Fonctions réseau potentielles qui peuvent être virtualisées
Tiré de Stallings (2015)

Network elements	**Functions**
Switching elements	Broadband network gateways, carrier grade NAT, routers
Mobile network nodes	Home location register, home subscriber server, gateway, General packet radio service (GPRS) protocol support node, radio network controller, various node B functions
Customer premises equipment	Home routers, set-top boxes
Tunneling gateway elements	IPsec/Secure sockets layer (SSL) virtual private network gateways
Traffic analysis	Deep packet inspection (DPI), QoE measurement
Assurance	Service assurance, Service level agreement (SLA) monitoring, testing and diagnostics
Signaling	Session border controllers, IP multimedia subsystem (IMS) components
Control plane/access functions	AAA (Authentication, authorization, and accounting) servers, policy control and charging platforms, Dynamic host configuration protocol (DHCP) servers
Application optimization	Content delivery networks, cache servers, load balancers, accelerators
Security	Firewalls, virus scanners, intrusion detection systems, spam protection

- Une meilleure flexibilité : la virtualisation des fonctions réseau accélère le déploiement des fonctionnalités réseau et permet d'effectuer des changements rapides afin de soutenir les nouveaux objectifs et produits organisationnels. Elle permet d'adapter rapidement le réseau aux fluctuations du trafic, à l'augmentation de la demande et aux nouveaux besoins ;
- Une meilleure efficacité : dans un contexte où la demande du réseau fluctue, la virtualisation des fonctions de réseau permet d'ajouter facilement des composantes et des fonctions réseau ainsi que sa topologie au lieu d'installer des équipements physiques. Elle permet également à plusieurs fonctions de s'exécuter sur un seul serveur, permettant ainsi la consolidation des ressources et la réduction à la fois des coûts d'investissements et des coûts opérationnels ;
- Élimination des équipements propriétaires : il coûte cher aux organisations à configurer et à déployer des équipements propriétaires qui peuvent facilement devenir obsolète. La

virtualisation des fonctions de réseau permet au matériel standard (par ex., serveurs) d'exécuter des fonctions réseau en remplaçant les équipements dédiés ;

- Modèle de paiement à l'utilisation : le paiement pour l'utilisation de ces fonctions réseau peut être effectué en fonction de l'utilisation des ressources (par ex., bande passante, CPU, mémoire et stockage).

1.6 Chaînes de service de réseau

Le réseau défini par logiciel et la virtualisation des fonctions réseau sont des technologies qui sont en train de révolutionner l'industrie de télécommunication et de l'internet puisqu'ils transforment l'architecture des réseaux ainsi que les stratégies pour fournir les services réseautiques. En effet, ils permettent de créer des réseaux personnalisés sur des infrastructures virtuelles. Comparativement aux réseaux traditionnels, ces nouveaux réseaux sont constitués d'une série de fonctions réseau virtuelles. La composition de ces fonctions dans un ordre particulier permettent de créer une chaîne de service de réseau (*Service Function Chain* - SFC).

Halpern et al. (Halpern, Pignataro et al., 2015) ont défini une chaîne de fonctions de service comme étant un ensemble ordonné de fonctions réseau virtuel abstraites avec des contraintes sur l'ordre des fonctions qui doivent être appliquées trafic. Ces fonctions virtuelles sont assignées à des machines virtuelles, conteneurs ou à des équipements physiques dédiés.

Avec les avancées de la virtualisation des fonctions réseau (NFV), les instances de services réseau sont fournies via plusieurs infonuages pour des raisons de performances et d'équilibrage de charge. L'interconnexion de ces instances pour former un service réseau de bout en bout complet est une tâche complexe, longue et coûteuse. Le chaînage de fonctions de service (SFC) est un mécanisme qui permet à différentes fonctions de service d'être connectées pour former un service. Le SFC permet de réaliser un gain économique et une meilleure flexibilité comparée aux environnements statiques (Bhamare, Jain, Samaka & Erbad, 2016).

La virtualisation des fonctions réseau et les chaines de service permettent aussi de réduire les pannes à l'aide l'automatisation des processus à travers l'infonuage puisque les interventions

humaines constituent une grande source d'erreur sur le réseau. Selon Juniper Networks inc. (Bednarz, 2016), l'humain est responsable de plus de 52% des pannes dans le réseau.

1.7 Les concepts de l'ordonnancement

1.7.1 Ordonnancement

L'ordonnancement est un aspect très important qui impacte considérablement la performance de l'environnement infonuagique. Ce processus pourra s'appliquer à différents niveaux de l'architecture infonuagique : niveau service, niveau tâche et niveau machine.

1. L'ordonnancement des services : l'infonuagique apparaît aujourd'hui de plus en plus adoptée dans de nombreux domaines en proposant plusieurs services. Dans ce cas, le principal critère à optimiser est le profit. Cependant, le type d'équipements proposés consomme une part d'énergie importante. Par conséquent, la planification en tenant compte de l'énergie est cruciale pour les systèmes à grande échelle qui consomment une quantité considérable d'énergie (Kessaci, 2013).
2. L'ordonnancement des tâches : l'objectif est d'optimiser l'assignation des tâches selon les contraintes de la QoS de chacun d'entre eux et de chaque client ainsi que de minimiser le coût total et le temps d'exécution.
3. L'ordonnancement des machines virtuelles : permets de fournir un ensemble des machines virtuelles demandées par l'ordonnancement des tâches. Le but est de trouver le meilleur ordonnancement des machines virtuelles sur les hôtes qui composent les centres de données dans un univers infonuagique (Femmam, Kazar, Baarir & Fareh, 2015).

1.7.2 Tâches

Les tâches : est une entité composée d'un ensemble d'opérations qui nécessite pour son exécution certaines ressources. Chaque tâche est définie par une date de début et une date de fin. Ces opérations peuvent être traitées indépendamment du flux principal des applications d'une manière asynchrone à l'aide d'un planificateur des tâches. Grâce aux technologies infonuagiques

les entreprises peuvent louer de la puissance de calcul sous forme de machines virtuelles aux utilisateurs (Fangzhe, Jennifer & Ramesh, 2010). Étant donné le grand nombre des machines virtuelles (VM) qui seront sollicitées, il est difficile de gérer manuellement l'attribution des tâches aux ressources informatiques dans le nuage (Huang & Huang, 2010). Nous avons donc besoin d'un planificateur des tâches qui implémente un algorithme efficace qui permettra d'optimiser leurs planifications dans l'environnement infonuagique. Un bon planificateur de tâches doit adapter sa stratégie de planification au changement de l'environnement et aux types de tâches (Gao, Wang & Xi, 2014).

1.7.3 Ressources

Dans un monde infonuagique, les clients peuvent utiliser des services offerts par un fournisseur de services infonuagiques. Ces services se basent sur des Machines Virtuelles (VM) et offrent par exemple des capacités de calcul et/ou de stockage spécifiques sous la forme d'instances infonuagiques. L'utilisation de ces instances dépend à la fois de leur disponibilité et du mécanisme de tarification appliqué par le fournisseur de services infonuagiques. Les ressources du nuage (par exemple, réseaux, serveurs, stockage, applications et services) sont utilisées à la demande, et peuvent être des ressources dédiées ou partagées.

1.7.4 Allocation des ressources

L'allocation de ressources est le processus de partage et de répartition optimale d'une quantité limitée des ressources disponibles selon des critères et des hypothèses définis. Dans le contexte infonuagique, l'allocation de ressources permettra aux fournisseurs de services de gérer les ressources allouées à chaque application selon des accords de niveau de service établis par négociation entre le fournisseur de services et les clients.

Dans ce sens, le placement des VNF dans l'infonuage a fait l'objet de plusieurs recherches. Leur objectif est principalement de minimiser le nombre d'instances de VNFs utilisées dans l'infonuage ainsi que le coût global du réseau en utilisant des algorithmes d'allocation de

ressources basés sur des heuristiques (Herrera & Botero, 2016; Lukovszki, Rost & Schmid, 2016; Ghaznavi, Shahriar, Ahmed & Boutaba, 2016; Bari, Chowdhury, Ahmed & Boutaba, 2015b; Mijumbi, Serrat, Gorricho, Bouten, De Turck & Davy, 2015; Gember, Akella, Anand, Benson & Grandl, 2012; Clayman, Maini, Galis, Manzalini & Mazzocca, 2014; Luizelli, Bays, Buriol, Barcellos & Gaspary, 2015; Rankothge, Le, Russo & Lobo, 2015a; Rankothge, Ma, Le, Russo & Lobo, 2015b; Cohen, Lewin-Eytan, Naor & Raz, 2015).

CHAPITRE 2

REVUE DE LA LITTÉRATURE

2.1 Introduction

Dans ce chapitre, nous passons en revue les approches existantes reliées aux différents objectifs de cette thèse. En particulier, on s'intéresse aux travaux reliés à l'approvisionnement des chaînes de fonctions de service et ensuite ceux reliés à la gestion du trafic dans les réseaux multi-domaines.

2.2 Approvisionnement des chaînes de fonctions de service

Dans cette section, nous présentons brièvement des travaux de recherche récents abordant le problème d'approvisionnement des chaînes de fonctions de service. Au cours des dernières années, un grand nombre de travaux ont abordé le problème de d'allocation de ressources pour les SFCs qui implique le placement et le chaînage de plusieurs VNFs (Ghaznavi, Khan, Shahriar, Alsubhi, Ahmed & Boutaba, 2015; Racheg, Ghrada & Zhani, 2017a; Carpio, Dhahri & Jukan, 2017; Wang, Wu, Le, Liu, Li & Lau, 2016; Alomari, Zhani, Aloqaily & Bouachir, 2020).

Plusieurs études de placement de VNFs ont visé à minimiser les coûts opérationnels et l'utilisation des liens Murukan, Jamaluddine, Kolhapure, Mikhael & Nouzari (2016); Zhang, Wu, Li & Lau (2017); Gupta, Samaka, Jain, Erbad, Bhamare & Metz (2017); Murukan *et al.* (2016); Ma, Beltran, Pan, Pan & Pissinou (2017), la minimisation de la latence Li, Hong, Xue & Pei (2019); Patel, Vutukuru & Krishnaswamy (2017); Yang, Zhang & Hong (2016); Bhamare, Samaka, Erbad, Jain, Gupta & Chan (2017); El Khoury, Ayoubi & Assi (2016); Krishnaswamy, Kothari & Gabale (2015); Kim, Han & Park (2016) ainsi que la minimisation de la consommation des ressources Tajiki, Salsano, Chiaraviglio, Shojafar & Akbari (2018); Xu, Zhang, Yu & Zhang (2018); Addis, Belabed, Bouet & Secci (2015); Yang *et al.* (2016); El Khoury *et al.* (2016); Bari, Chowdhury, Ahmed, Boutaba & Duarte (2016); Kim *et al.* (2016); Pham, Tran, Ren, Saad & Hong (2017).

L'analyse des objectifs de placement des SFCs par Santos et al. Santos, Bezerra, Rocha, Ferreira, Moreira, Gonçalves, Marquezini, Recse, Mehta, Kelner et al. (2022), ont trouvé que la minimisation des coûts opérationnels a été l'objectif le plus recherché avec environ 42% des articles publiés.

Par exemple, Carpio et al. (Carpio *et al.*, 2017) modélisent le placement et le chaînage des VNFs en tant que programme linéaire mixte (MIP) et le comparent à un algorithme de placement aléatoire. Pour une meilleure extensibilité, ils conçoivent un algorithme génétique pour trouver une solution sous-optimale. L'algorithme proposé se concentre d'abord sur la recherche de chemins admissibles et le calcul des coûts de liaison, puis sur l'allocation au sein de ces chemins des ressources pour les VNFs. Cependant, l'algorithme suppose que le nombre d'instances de chaque VNF est connu à l'avance.

Bari et al. (Bari, Chowdhury, Ahmed & Boutaba, 2015a) ont défini un système de fichiers Linux et des fonctionnalités qui peuvent être exploitées pour implémenter certaines fonctions réseau. Cependant, la méthode proposée ne traite pas le cas où plusieurs instances sont créées et ne détermine pas le placement approprié des VNFs dans l'infrastructure. Beck et Botero (Beck & Botero, 2015) ont également examiné le problème de placement et de chaînage VNF et ont proposé CoordVNF, une solution qui vise à minimiser l'utilisation des liens réseau de l'infrastructure. Cette proposition ne prend pas en compte plusieurs instances du même VNF.

Wang et al. (Wang *et al.*, 2016) attaque le problème de déploiement en ligne de plusieurs instances de VNFs afin de traiter le trafic fluctuant reçu au niveau des VNFs avec l'objectif de réduire le coût minimum d'approvisionnement des ressources. Ghaznavi et al. Ghaznavi *et al.* (2015) abordent le placement Elastic VNF dans le but de réduire la consommation de serveur et de la bande passante. Ils introduisent ainsi une solution pour optimiser le placement VNF en minimisant les coûts d'installation, de transport, de réaffectation et de migration des instances de VNFs. Cependant, la solution suppose que toutes les instances sont du même type.

Contrairement aux travaux précédents, nous considérons non seulement le mappage des SFCs mais également la phase de traduction où le nombre d'instances pour chaque VNF est estimé et

considéré pour construire un réseau virtuel à mapper. Nous abordons également le mappage de SFCs en tenant compte des coûts de synchronisation et de déploiement des instances de VNF. Nous nous appuyons également sur une étude qu'on réalise basées sur les coûts des instances d'Amazon EC2.

2.3 Gestion du trafic dans les réseaux multi-domaines

Plusieurs efforts de recherche récents ont proposé différentes formes de collaboration entre des domaines multiples afin d'assurer une gestion plus efficace du trafic dans les réseaux multi-domaines. Ainsi, nous avons identifié trois catégories de collaboration : la collaboration basée sur des protocoles, la collaboration contrôleur-à-contrôleur (C-à-C) et la collaboration basée sur un courtier. Dans ce qui suit, nous fournissons plus de détails sur chacune de ces catégories et analysons la littérature existante et de ses limites.

2.3.1 Collaboration basée sur des protocoles

La collaboration basée sur des protocoles fait référence à la collaboration entre différents domaines via un protocole qui effectue une telle collaboration en échangeant des informations entre les domaines. L'exemple typique et le plus courant de tels protocoles est le Border Gateway Protocol (BGP) (Manolova & Ruepp, 2010; Giorgetti, Paolucci, Cugini & Castoldi, 2011). BGP est un protocole de routage interdomaine qui effectue l'échange d'informations de routage et d'accessibilité entre les domaines. Cependant, BGP ne prend pas en compte les mesures en temps réel (par exemple, la bande passante, la latence) et l'existence de multiples routes possibles dans les domaines traversés. En conséquence, avec le BGP, le trafic peut être acheminé via un domaine même s'il est congestionné (Siracusa, Grita, Maier, Pattavina, Paolucci, Cugini & Castoldi, 2012). De plus, comme le BGP est décentralisé, cela conduit à un temps de convergence assez long après chaque changement de topologie. De plus, la restauration des tables de routage BGP après un tel changement peut prendre entre 3 et 15 minutes (Labovitz, Ahuja, Bose & Jahanian, 2000). Ce délai nuit considérablement aux performances, en particulier pour les applications sensibles au délai (Nayyer, Sharma & Awasthi, 2019).

Pour remédier à ces limites, Kotronis et al. (Kotronis, Dimitropoulos & Ager, 2012; Kotronis, Gämperli & Dimitropoulos, 2015) propose une approche hybride BGP-SDN avec un plan de contrôle logiquement centralisé pour plusieurs domaines. Les domaines doivent externaliser leur logique de contrôle de routage interdomaine vers un contrôleur SDN externe. Cependant, Thai et al. (Thai & de Oliveira, 2013) a démontré que BGP n'est pas approprié pour le réseau SDN de nouvelle génération, en raison de la capacité limitée du protocole à contrôler le réseau.

Feamster et al. Feamster, Rexford, Shenker, Clark, Hutchins, Levin & Bailey (2013) ont proposé un échange Internet défini par logiciel (SDX) qui combine les fonctionnalités SDN avec le routage inter-domaine BGP. Ce travail vise à tirer parti des points d'échange Internet (IXP). Cependant, cette approche est réservée uniquement aux domaines connectés aux IXP. De plus, comme les IXPs reposent sur BGP, lorsque la topologie change, il faut beaucoup de temps pour converger et récupérer les nouvelles routes Labovitz, Ahuja, Bose & Jahanian (2001).

L'étude du déploiement hybride des réseaux SDN et IP menée par Lin et al. Lin, Hart, Krishnaswamy, Murakami, Kobayashi, Al-Shabibi, Wang & Bi (2013) a proposé le déploiement incrémental de SDN. Cependant, cette solution suggère d'appliquer le protocole BGP entre les domaines SDN et IP ou entre les domaines SDN pour assurer la connectivité sans vraiment viser à améliorer la performance du routage inter-domaine ou apporter des garanties sur les délais de bout en bout.

Douville et al. Douville, Rougier, Secci et al. (2008) a proposé une architecture permettant le approvisionnement automatique des inter-AS GMPLS-TE, basée sur l'introduction d'un plan de service multidomaine couplé à l'architecture à base de PCE. Ce travail a introduit un plan de service dédié à l'échange d'informations inter-domaines. Pour permettre la composition d'un service de réseau de bout en bout, chaque opérateur annonce aux autres domaines ses éléments de service. Cependant, la propagation de nouvelles informations dans plusieurs domaines lorsqu'un changement se produit prendrait du temps par rapport à la propagation des informations vers un seul courtier centralisé.

Casellas et al. Casellas, Martínez, Muñoz, Vilalta & Liu (2015a) a proposé un mécanisme d'orchestration distribuée qui utilise une approche hybride, combinant des éléments de contrôle distribué avec des éléments de contrôle centralisé. Ils proposent une architecture de contrôle multi-domaine organisée en couches, où les contrôleurs SDN utilisent les protocoles GMPLS comme interfaces Est et Ouest. Cependant, ces architectures distribuées nécessitent des interactions complexes entre les contrôleurs pour mettre à jour la topologie et augmente le temps de convergence nécessaire pour synchroniser l'état du réseau entre les domaines Choi & Li (2016).

Selon Elguea et al. (Elguea & Martinez-Rios, 2019, 2017), les protocoles de routage doivent optimiser le calcul des chemins en fonction de la latence afin de minimiser le nombre de sauts entre les domaines pour atteindre une destination. Ils proposent donc de mesurer la latence de plusieurs routes sur plusieurs domaines puis de les comparer à l'aide de méthodes statistiques. Une fois la route optimale calculée, elle est injectée dans les routeurs BGP. Cependant, comme chaque domaine n'a pas le contrôle sur les autres, le chemin interne du domaine peut changer et les performances ne sont donc pas garanties.

2.3.2 Collaboration contrôleur-à-contrôleur

La collaboration contrôleur-à-contrôleur (C-à-C) fait référence à la collaboration entre plusieurs domaines SDN où les contrôleurs échangent des informations afin d'optimiser les décisions de routage. Par exemple, Tootoonchian et Ganjali (Tootoonchian & Ganjali, 2010) proposent une approche de plan de contrôle distribué appelée HyperFlow. HyperFlow permet de distribuer le plan de contrôle en répliquant l'état du contrôleur via des mises à jour périodiques afin de conserver la même vue réseau sur tous les contrôleurs. RMOF (Phan, Thoai & Kuonen, 2013) est une architecture collaboratrice où plusieurs contrôleurs collaborent pour maintenir un réseau mondial. Le système d'exploitation de réseau ouvert (ONOS) (Berde, Gerola, Hart, Higuchi, Kobayashi, Koide, Lantz, O'Connor, Radoslavov, Snow & Parulkar, 2014) utilise également des contrôleurs distribués fonctionnant ensemble pour fournir une vue globale du réseau, facilitant des fonctionnalités avancées telles que l'équilibrage de charge, la tolérance aux pannes et la sécurité. Le multicontrôleur OpenDaylight (ODL) utilise des APIs orientés

East-West chargés de collecter et de partager les informations entre les contrôleurs fédérés (Medved, Varga, Tkacik & Gray, 2014; Jahan, Shaik, Kotaru, Sangam & Kuppili, 2018; Yin, Xie, Tsou, Lopez, Aranda & Sidi, 2012). RMOF, ONOS et ODL supposent que tous les contrôleurs ont exactement la même vue et, par conséquent, ils ne sont pas adaptés aux scénarios où seule une information limitée est partagée entre les différents domaines administratifs impliqués.

Karakus et coll. (Karakus & Durresi, 2015) ont proposé une architecture de réseau hiérarchique composée d'un courtier constitué d'un super contrôleur et de plusieurs domaines SDN avec chacun ayant un contrôleur local. Leurs expériences démontrent comment cette architecture hiérarchique peut gérer plus de trafic que celle non hiérarchique avec plusieurs contrôleurs.

Gerola et coll. (Gerola, Lucrezia, Santuari, Salvadori, Ventre, Salsano & Campanella, 2016) ont proposé Inter Cluster Onos Network Application (ICONA), une application qui permet d'interconnecter plusieurs clusters ONOS pour répartir la charge du plan de contrôle entre leurs contrôleurs. ICONA a implémenté une interface de communication est-ouest pour permettre la communication entre tous les clusters.

Kalpana et coll. (Joshi & Kataoka, 2019) a proposé PRIME-Q, un nouveau mécanisme pour calculer le chemin de bout en bout à travers un réseau SDN multi-domaine basé sur des contraintes de qualité de service (*Quality of Service* - QoS). Avec PRIME-Q, le contrôleur du domaine source envoie la demande QoS (par exemple, délai, bande passante) a tous les domaines croisés et obtient une décision d'acceptation ou de rejet de chacun d'eux avant de prendre la décision finale. Cependant, le manque de visibilité des topologies des domaines peut rendre difficile la recherche du meilleur chemin de bout en bout pour le trafic. De plus, cette approche implique de nombreux échanges pour négocier la QoS entre tous les domaines croisés, ce qui nécessite plus de temps.

Un autre travail pertinent est celui de Lin et al. Lin, Bi, Wolff, Wang, Xu, Chen, Hu & Lin (2015) qui a proposé un nouveau mécanisme de routage inter-domaine nommé *West-East bridge* pour permettre à différents domaines administratifs SDN de coopérer. Le pont Ouest-Est est une plate-forme pour échanger les informations du réseau entre différents domaines SDN. Chaque

contrôleur de domaine administratif SDN construit une vue globale du réseau basée sur toutes les informations partagées avec d'autres domaines. Comme chaque contrôleur de domaine construit sa propre vue globale du réseau, cela peut entraîner un long temps de convergence après chaque changement de topologie et peut créer une discordance entre les vues globales du réseau dans les différents domaines.

2.3.3 Collaboration basée sur un courtier

La collaboration basée sur un courtier entre plusieurs domaines SDN repose sur une composante centrale appelée courtier. Le courtier collecte les données de la part des différents contrôleurs de domaine et assure la collaboration entre eux.

Farrel et coll.(Farrel, Vasseur & Ash, 2006b; King & Farrel, 2012) ont proposé d'utiliser une composante centrale appelée Hierarchical Path Computation Element (H-PCE) pour calculer les chemins de bout en bout pour les réseaux multi-domaines. H-PCE utilise un PCE parent unique qui coordonne les PCE enfants situés dans les domaines. Le PCE-parent est en charge du calcul des chemins inter-domaines tandis que chaque PCE enfant s'occupe du calcul des chemins dans son propre domaine.

H-PCE utilise BGP pour récupérer les informations des domaines utilisées dans le calcul du chemin de bout en bout et hérite donc des limitations BGP. Il ne prend pas en compte les mesures en temps réel (par exemple, bande passante, latence) et l'existence de multiples routes possibles dans les domaines traversés (Siracusa *et al.*, 2012).

Giorgetti Giorgetti (2015) et Casellas et al. Casellas, Muñoz, Martínez, Vilalta, Liu, Tsuritani, Morita, López, de Dios & Fernández-Palacios (2015b) ont proposé une architecture d'orchestration hiérarchique basée sur le PCE où le pPCE maintient la topologie inter-domaine abstraite à travers les cPCE via le protocole Border Gateway Protocol-Link State (BGP-LS) ou le protocole de communication PCE. Cette approche réduit le temps de convergence nécessaire pour synchroniser l'état du réseau et la charge de trafic de contrôle pour les rafraîchissements d'état périodiques.

Tableau 2.1 Résumé des travaux sur la collaboration multi-domaine

Référence	**Catégorie de collaboration**	**Vue du contrôleur de domaine**	**Informations partagées (entre domaine)**
(Manolova & Ruepp, 2010; Giorgetti *et al.*, 2011)	Basée sur protocole	Chemins inter-domaines (liste ordonnée d'ID de domaines)	Poids des chemins inter-domaines
(Elguea & Martinez-Rios, 2019, 2017)	Basée sur protocole	Chemins inter-domaines (liste ordonnée d'ID de domaines)	Latence de bout en bout
(Kotronis *et al.*, 2012, 2015)	Basée sur protocole	Chemins inter-domaines (liste ordonnée d'ID de domaines)	Poids des chemins inter-domaines
(Tootoonchian & Ganjali, 2010)	C-à-C	Liens inter et intra-domaines	Bande passante
(Phan *et al.*, 2013)	C-à-C	Liens inter et intra-domaines	Coût de routage entre les commutateurs
(Medved *et al.*, 2014; Jahan *et al.*, 2018; Yin *et al.*, 2012)	C-à-C	Liens inter et intra-domaines	Tous les informations
(Gerola *et al.*, 2016)	C-à-C	Liens inter et intra-domaines	Délai du lien, bande passante et nombre de flux par lien
(Joshi & Kataoka, 2019)	C-à-C	Chemins inter-domaines (liste ordonnée d'ID de domaines)	Décision binaire basée sur une requête QoS
(Vilalta, Mayoral, Pubill, Casellas, Martínez, Serra, Verikoukis & Muñoz, 2016)	Basée sur courtier	Liens inter et intra-domaines	Tous les informations
(Marconett & Yoo, 2015)	Basée sur courtier	Liens inter et intra-domaines	Capacité du lien, utilisation du lien, délai, taux de perte
(Nayyer *et al.*, 2019)	Basée sur courtier	Liens inter et intra-domaines	Tous les informations
(Karakus & Durresi, 2015)	Basée sur courtier	Inter and intra-domain paths	Bande passante des inter et intra-chemins
Notre approche	Basée sur courtier	Chemins interdomaines et informations limitées sur les chemins intra-domaine	Bande passante, utilisation, retard, perte

La procédure de calcul rétro-récursif basé sur PCE (BRPC) est une technique de calcul de chemin inter-PCE Farrel, Vasseur & Ash (2006a); Vasseur, Zhang, Bitar & Le Roux (2009). La procédure BRPC est basée sur le protocole PCE et permet le calcul synchronisé des chemins à commutation d'étiquettes avec les exigences de qualité de service demandées. BRPC calcule récursivement, à chaque PCE de chacun des domaines, un arbre virtuel du plus court chemin (VSPT). La limitation de cette approche est que la navigation à travers les domaines est complexe, car chaque système autonome peut recevoir plusieurs fois un VSPT partiel. Il est également difficile de calculer le chemin en l'absence d'une séquence de systèmes autonomes à l'avance.

FlowBroker (Marconett & Yoo, 2015) et Laman (Nayyer *et al.*, 2019) ont proposé une architecture hiérarchique basée sur un contrôleur de type courtier au sommet de tous les contrôleurs de domaine pour partager des données entre eux. Les données partagées sont la capacité, l'utilisation, le délai et le taux de perte de paquets.

2.3.4 Discussion

Le tableau 2.1 résume les travaux de recherche sur la collaboration multi-domaine et fournit :

1. la manière dont la collaboration est menée entre les différents domaines (basée sur des protocoles, C-à-C ou basée sur des courtiers),
2. la vue réseau de l'entité prenant les décisions de routage (c'est-à-dire, courtier, contrôleur ou nœud),
3. les informations partagées entre les domaines. La collaboration inter-domaine SDN continue d'être un défi pour l'échange de trafic entre les domaines du réseau, car il n'y a toujours pas de consensus autour d'une solution pouvant être déployée avec succès à grande échelle.

Contrairement aux solutions existantes, ce travail propose une nouvelle solution basée sur un courtier qui n'exploite que des informations partagées limitées, c'est-à-dire uniquement l'état des chemins disponibles (pas des liens) au sein de chaque domaine, afin de maximiser le nombre de flux desservis dans tous les domaines et de satisfaire leurs délais de bout en bout, leurs bandes passantes et leurs exigences en matière de coûts. A notre connaissance, ce travail est le premier

à envisager de maximiser le nombre de flux, tout en prenant en compte à la fois les exigences de délai, de bande passante et de coût.

CHAPITRE 3

SFCAAS : CHAÎNAGE DE FONCTION DE SERVICES EN TANT QUE SERVICE DANS LES ENVIRONNEMENTS VIRTUALISÉS

3.1 Introduction

Propulsée par les récentes avancées dans les technologies de virtualisation, la virtualisation des fonctions réseau (*Network Function Virtualization* - NFV), la réseautique définie par logiciel (*Software Defined Networking* - SDN) et la programmabilité du plan de données, la 'softwarization' du réseau est devenue une nouvelle tendance pour séparer le plan matériel du réseau du plan logiciel et implémenter les fonctions du réseau sous forme de logiciel (*software*). Cette tendance révolutionne la façon dont les réseaux sont conçus et gérés. Elle ouvre aussi la porte à un nouvel écosystème informatique ouvert avec de nouvelles plateformes, logiciels et applications réseau (Zhani & Elbakoury, 2020; Clemm, Zhani & Boutaba, 2020; Varghese, Leitner, Ray, Chard, Barker, Elkhatib, Herry, Hong, Singer, Tso, Yoneki & Zhani, 2019).

Les avantages de la *'softwarization'* du réseau sont nombreux. En effet, elle offre une meilleure agilité et flexibilité, une automatisation facile et une mise sur le marché plus rapide des services informatiques tout en garantissant leur évolutivité et en minimisant les dépenses d'investissement et d'exploitation.

La *'softwarization'* du réseau permet aux fournisseurs de réseau/cloud de proposer une nouvelle offre de services : les chaînes de fonctions de service (*Service Function Chains* - SFCs) en tant que service (SFCaaS) où une chaîne de fonctions de service pourrait être proposée en tant que service à un tiers. Un SFC est un ensemble ordonné de fonctions de réseau virtuel (*Virtual Network Function* - VNF) à traverser par les paquets arrivants (Figure 3.1). Les VNFs peuvent être des fonctions de réseau (par ex., pare-feu, IDS) implémentées dans une machine virtuelle ou un conteneur s'exécutant sur un serveur de base ou aussi implémentées dans un matériel dédié comme les cartes programmables (par ex., *Field-Programmable Gate Arrays* - FPGA).

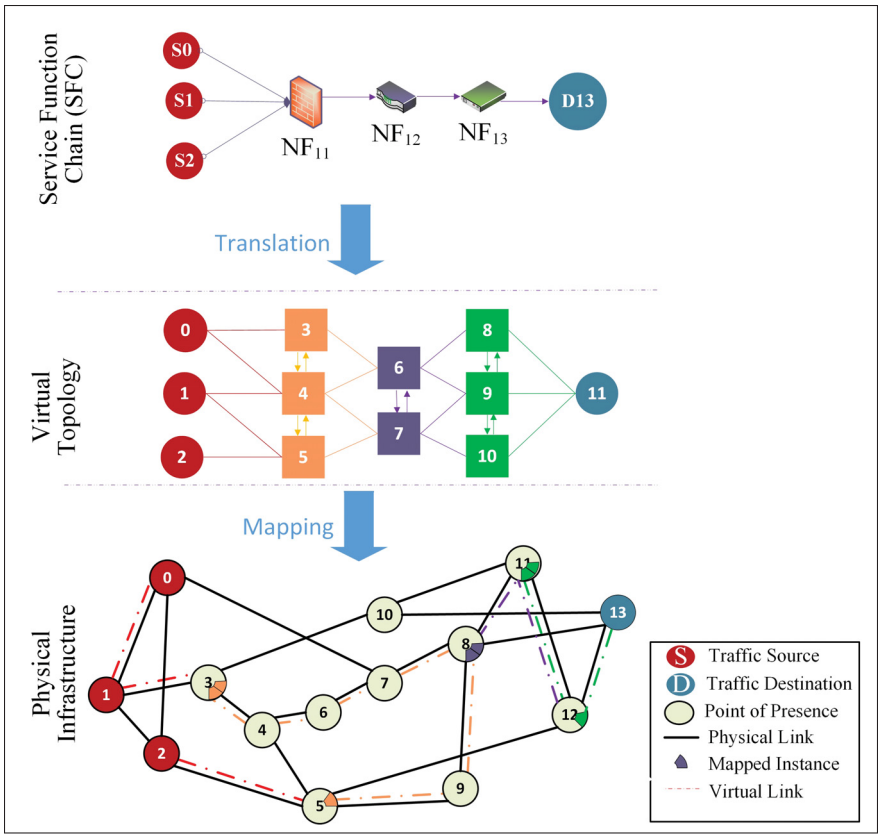


Figure 3.1 SFCaaS - Traduction et mappage SFC
Tiré de Zhani & Elbakoury (2020)

Dans ce contexte, un fournisseur de SFCs serait confronté à un défi majeur consistant à la manière d'allouer les ressources aux SFCs demandés dans l'infrastructure physique. Bien que ce problème ait été récemment largement abordé (Racheg, Ghrada & Zhani, 2017b; Tashtarian, Zhani, Fatemipour & Yazdani, 2020; Amokrane, Zhani, Langar, Boutaba & Pujolle, 2013) dans la littérature, nous revenons sur ce problème avec les nouvelles contributions suivantes :

1. Nous considérons deux phases pour approvisionner un SFC (Figure 3.1). La première est la phase de traduction (*translation phase*) qui vise à identifier le nombre optimal d'instances (c.-à.-d. machines virtuelles) qui sont nécessaires pour implémenter chaque VNF afin de répondre à la demande. La deuxième phase est la phase de mappage (*mapping phase*) visant à décider où placer les instances en tenant compte du coût de déploiement qui varie

considérablement d'un emplacement à un autre. À notre connaissance, aucun travail existant n'a considéré la phase de traduction.

2. Nous effectuons une étude détaillée des coûts des machines virtuelles (c.-à.-d. instances) offertes par Amazon EC2 (Amazon) par rapport à l'emplacement, la taille de l'instance et les performances. Cette étude de coût étend celle réalisée par les auteurs dans (Ghrada *et al.*, 2018) et permet principalement d'avoir des indicateurs clairs pour prendre les décisions sur le type, l'emplacement et le coût des instances.
3. Nous formulons le problème de mappage des machines virtuelles en tant que programme linéaire entier visant à réduire les coûts opérationnels des instances et des liens du fournisseur SFC ainsi que les coûts de synchronisation entre les instances de même type. Il convient de noter que les coûts de synchronisation n'ont pas été pris en compte dans la littérature existante.
4. Nous proposons deux algorithmes heuristiques pour résoudre le problème de mappage avec les mêmes objectifs mentionnés précédemment en tenant compte de l'étude des coûts des instances Amazon EC2. Le premier est un algorithme de base intuitif (appelé : 'Baseline') et le second est un algorithme plus sophistiqué d'approvisionnement basé sur la composition du SFC (appelé : 'SPIN') et qui fournit des résultats considérablement meilleur par rapport au Baseline.

Ce chapitre est organisé comme suit. La section 3.2 décrit le SFC en tant que modèle de service, mettant en évidence le modèle d'affaire et les parties prenantes potentielles impliquées ainsi que les défis techniques liés au déploiement d'un tel modèle. La section 3.3 présente l'étude détaillée des coûts des instances Amazon EC2 par rapport à plusieurs paramètres et présente les principaux résultats trouvés. La section 3.4 présente la formulation ILP du problème de mappage. Ensuite, la section 3.5 présente les deux algorithmes heuristiques pour résoudre le problème de mappage. Les résultats expérimentaux sont fournis dans la section 3.6. Enfin, les conclusions sont résumées dans la section 3.7.

3.2 SFCaaS - Modèle d'affaire, avantages et défis

Le modèle d'affaire présenté ci-dessus identifie les parties prenantes impliquées dans un environnement où les SFCs sont fournis en tant que service (SFCaaS). Nous identifions principalement trois parties prenantes définies comme suit :

- Fournisseur de SFCs : c'est une entreprise ou un fournisseur qui propose des «chaînes de fonctions de service» en tant que service (SFCaaS). Il possède et gère une infrastructure physique et est responsable du déploiement des plates-formes et des logiciels nécessaires à l'exécution des fonctions réseau et de l'approvisionnement et de la gestion des SFCs demandées. Le fournisseur de SFCs assure donc la phase de traduction des SFCs pour générer un réseau ou une topologie virtuelle, composée de machines virtuelles et de liens, qui doit être mappée sur l'infrastructure. Le réseau virtuel est obtenu en identifiant le nombre d'instances nécessaires pour implémenter chaque VNF et les liens virtuels utilisés pour les connecter (Figure 3.1). On note aussi que des liens virtuels doivent également être créés pour connecter les instances implémentant le même VNF afin d'assurer la synchronisation entre elles. En effet, la synchronisation est nécessaire pour garantir le fonctionnement normal de certaines fonctions réseau (par ex., IDS) (Alomari *et al.*, 2020).
- Fournisseur de services : il peut s'agir d'une entreprise ou d'une institution dont les utilisateurs sont répartis dans le monde entier. Un fournisseur de services doit définir le SFC requis, sa composition, les exigences de performance et identifier les sources / destinations de la chaîne. La composition du SFC fait référence au type de fonctions réseau composant la chaîne. Les exigences de performance pourraient être exprimées en termes de délai de bout en bout de la chaîne, de taux de perte de paquets, de la demande de trafic et d'autres paramètres. Bien entendu, le fournisseur de services compte sur le fournisseur de SFCs pour fournir le SFC et allouer les ressources nécessaires.
- Utilisateurs : les utilisateurs sont les clients d'un fournisseur de services (par exemple, les utilisateurs finaux) dont le trafic devra être dirigé à travers le SFC du fournisseur de services.

Les fournisseurs potentiels de SFCs pourraient être de grandes entreprises offrant des services infonuagiques comme Google, Amazon EC2, Microsoft et disposant de leur propre réseau qui permettrait d'avoir des performances prévisibles (Amazon, 2021; Google, 2021).

Par exemple, l'infrastructure AWS illustrée à la Figure 3.2 est une infrastructure mondiale définie par logiciel (Amazon, 2021) avec 25 régions à travers le monde desservant 245 pays et territoires où chaque région contient un ou plusieurs centres de données avec 218 emplacements périphériques et 12 caches périphériques régionaux pour un total de 230 points de présence (*Point Of Presence* - PoP). Les régions sont connectées via un réseau privé mondial géré par Amazon AWS.

Étant donné que les ressources informatiques sont disponibles partout dans cette infrastructure mondiale et qu'une telle entreprise possède l'expertise sur des technologies telles que l'infonuagique, la virtualisation, le SDN, il est simple de pouvoir fournir des SFCs en tant que service et de fournir et d'allouer facilement les ressources requises à travers leur infrastructure mondiale.

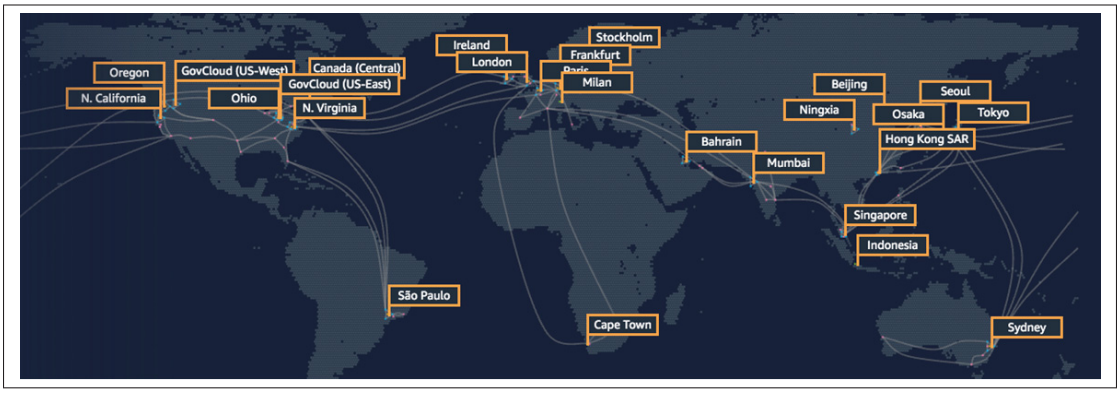


Figure 3.2 L'infrastructure cloud mondiale d'AWS
Tiré de Amazon (2021)

• **Avantages de SFC :** Similairement aux services infonuagiques traditionnels, offrir un service SFC apporterait plusieurs avantages aux fournisseurs de services, notamment en évitant les complications de gestion et en évitant le besoin de maintenance logicielle et matérielle. En outre, les coûts seront réduits, car il n'y a pas de dépenses d'investissement et de fonctionnement et il est possible d'obtenir des bas prix grâce aux économies d'échelle. De plus, le SFCaaS permettrait

de meilleures performances grâce à l'expertise et la maîtrise totale de l'infrastructure par le fournisseur de SFCs (c.-à-d. Topologies, caractéristiques, performances). Les fournisseurs de SFCs pourraient également proposer des fonctions réseau personnalisées qui sont soigneusement mises en œuvre et gérées pour offrir des performances optimales.

• **Défis techniques :** Du point de vue du fournisseur de SFCs, le principal défi est de fournir des SFCs avec l'objectif de maximiser les profits, de minimiser les coûts opérationnels tout en satisfaisant les exigences des SFCs en termes de délais de bout en bout et bande passante. Dans ce contexte, nous pouvons identifier plusieurs défis qui peuvent être résumés comme suit :

- Décider du nombre d'instances à utiliser pour implémenter un VNF : implémenter le même VNF dans plusieurs instances permet de surmonter le manque de ressources en répartissant la fonction entre plusieurs points de présence. Elle permet également de réduire les coûts et d'améliorer la tolérance aux pannes, car la fonction ne sera pas interrompue en cas de défaillance puisque d'autres instances prendront le relais. D'un autre côté, la mise en œuvre du même VNF dans plusieurs instances présente plusieurs inconvénients, notamment le coût de la synchronisation entre les instances qui peut être traduit par une consommation de CPU, de mémoire et de bande passante. De plus, il serait nécessaire d'imposer des limites de délais sévères pour effectuer la synchronisation entre les instances pour assurer le bon fonctionnement de la fonction réseau implémentée.
- Décider du type d'instance de VM à utiliser pour exécuter le VNF : la sélection du type d'instance de VM dépend des exigences de la fonction réseau en termes de ressources (c.-à-d. vCPU, mémoire, stockage), de capacité de traitement (paquets par seconde), et de coût opérationnel relié à l'exécution de l'instance. Bien entendu, cette sélection doit prendre en compte les propriétés de la fonction réseau (la fonction elle-même, le logiciel, le système d'exploitation, la base de données et autres services) et son emplacement géographique qui a un impact sur le coût de l'instance et le délai d'accès.
- Placer et chaîner des instances : le troisième défi est d'identifier où placer les instances et comment allouer les ressources en bande passante pour les liens virtuels pour chaîner ces instances.

Dans ce chapitre, nous abordons les défis mentionnés ci-dessus et étudions les paramètres et les considérations connexes pour les traiter. Dans ce qui suit, nous commençons par une revue de la littérature qui a abordé ces mêmes défis.

3.3 Étude des coûts des instances Amazon EC2

Dans cette section, nous étudions les coûts (prix) des instances de machine virtuelle Amazon EC2 en fonction des ressources, de l'emplacement et des performances. Cette étude de coût étend celle réalisée par les auteurs dans (Ghrada *et al.*, 2018) et permet principalement d'avoir des indicateurs clairs pour prendre les décisions sur le type, l'emplacement et le coût des instances.

Plus précisément, nous avons considéré les instances (machine virtuelle) à usage général d'Amazon EC2 de type T2 (Amazon) qui fournissent une large gamme d'offres, comme indiqué dans la Figure 3.3. Ces instances à usage général offrent des ressources de calcul, de mémoire et de réseau qui peuvent être utilisées pour divers types de charges de travail, y compris les fonctions réseau. Chaque type d'instance définit la quantité de vCPU et de mémoire de la machine virtuelle selon des coûts différents. Le tableau montre les caractéristiques des serveurs sur lesquels les VMs seront exécutées (Figure 3.3).

Dans ce qui suit, nous étudions plusieurs aspects liés au coût des différentes instances et à leurs performances. L'étude comprend le coût de l'instance par rapport à l'emplacement, le coût par rapport à la pile logicielle, la quantité de ressource de l'instance par rapport au coût et enfin, les performances du VNF (c'est-à-dire la capacité de traitement des paquets) par rapport au type d'instance.

• **Coût de l'instance par rapport à l'emplacement :** La Figure 3.4 montre le prix de l'instance pour 15 emplacements dans l'infrastructure Amazon. Il est clair que les coûts des instances varient considérablement d'un endroit à l'autre. Il convient de noter qu'il y a un impact élevé, même pour une petite différence dans le coût de l'instance. Par exemple, une différence de coût de 0,1 $/heure se traduirait par 86 millions de dollars par an pour 100K instances, et environ 2 milliards de dollars pour environ 2 millions d'instances. En effet, deux millions d'instances

Instance Type	vCPU Cores	Memory (GiB)	Physical machine Processor
t2.micro	1	1	High frequency Intel Xeon processors
t2.small	1	2	
t2.medium	2	4	
t2.large	2	8	
t2.xlarge	4	16	
t2.2xlarge	8	32	
m4.4xlarge	16	64	2.3 GHz Intel Xeon® E5-2686 v4
m4.10xlarge	40	160	
m4.16xlarge	64	256	2.4 GHz Intel Xeon® E5-2676 v3

Figure 3.3 Instances à usage général EC2
Tiré de Amazon

est une estimation de la limite inférieure du nombre d'instances exécutées sur l'infrastructure Amazon EC2 (Amazon, last accessed May 2021).

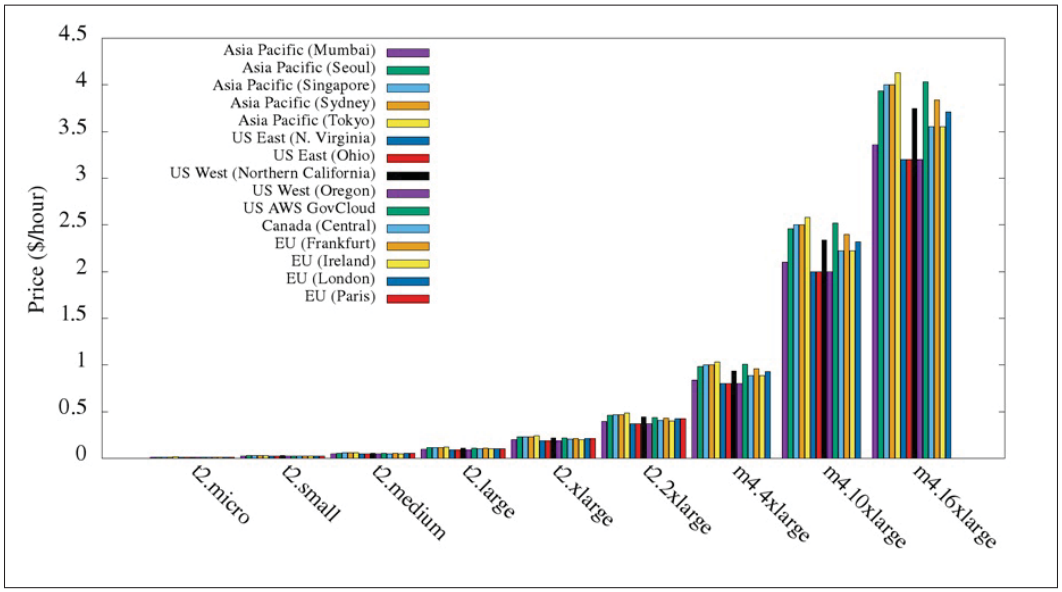


Figure 3.4 Prix de l'instance pour différents emplacements

• **Coût par rapport à la pile de logiciels :** La Figure 3.5 montre le prix de différentes instances avec différente pile logicielle. Il ne prend en compte que les instances situées dans la région Oregon Amazon AWS. La figure montre que les coûts d'instance varient en fonction des piles logicielles. Les distributions Linux (par exemple, Linux, RHEL, SLES) ont des coûts similaires et sont beaucoup moins chères que les instances exécutant Microsoft Windows. De plus, l'ajout de logiciels supplémentaires aux instances (par exemple, SQL Web) augmenterait considérablement le prix.

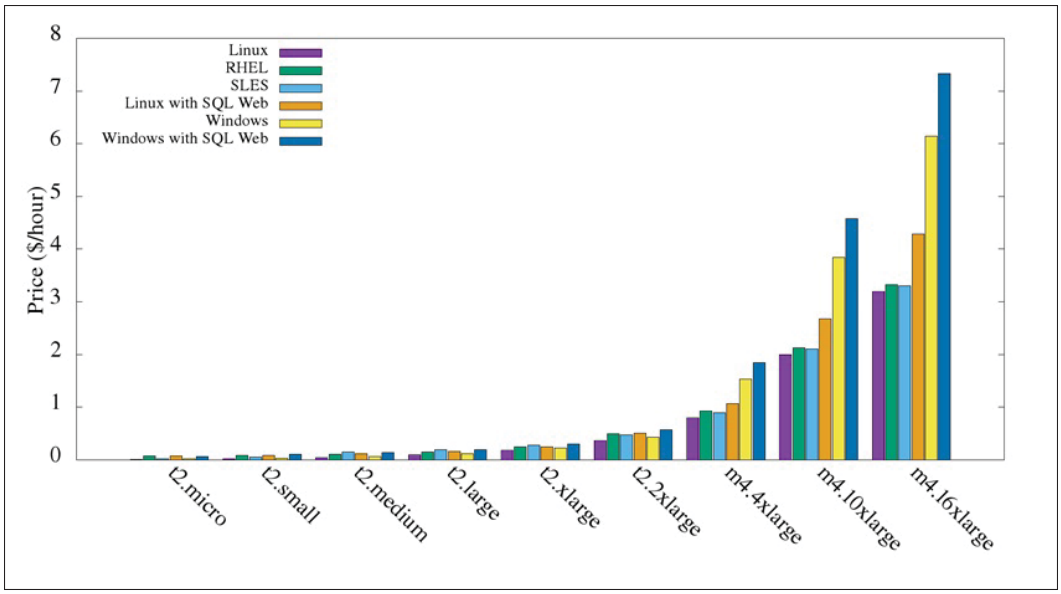


Figure 3.5 Prix de l'instance pour différentes piles de logiciels (Oregon)

• **Taille de l'instance vs coût** : La taille de l'instance fait référence à la quantité de ressources en termes de CPU et de mémoire dont dispose une instance. Nous visons donc à évaluer, pour le même coût, la quantité de ressources que nous pouvons obtenir lorsque nous utilisons des micro-instances par rapport à des instances plus grandes. Pour ce faire, la Figure 3.6 (extraite de Ghrada *et al.* (2018)) montre le prix de toutes les instances AWS et la quantité de ressources qu'elles fournissent. Elle montre également combien d'instances t2.micro (contenant 1 vCPU et 1 GiB de mémoire) pourraient être approvisionnées pour le même prix.

Par exemple, si on considère le prix d'une instance m3.16xlarge (contenant 64 vCPU et 256 GiB de mémoire), il est égal à 3,2 $/heure. Pour presque le même prix, on pourrait approvisionner 256 instances t2.micro offrant 256 vCPU et 256 GiB de mémoire. Cela signifie que si on approvisionne 256 instances t2.micro, nous pouvons obtenir 192 (soit 256-64) processeurs virtuels supplémentaires avec la même quantité de mémoire (256 GiB) par rapport à une seule instance m3.16xlarge. La même remarque s'applique aux autres types d'instances.

Par conséquent, nous pouvons conclure que les petites instances sont plus rentables que les grandes instances, car, pour le même coût, les micro-instances fourniraient environ quatre fois plus de processeurs virtuels. Cela est bien sûr intéressant si la fonction/application pouvait fonctionner normalement de manière répartie sur plusieurs instances.

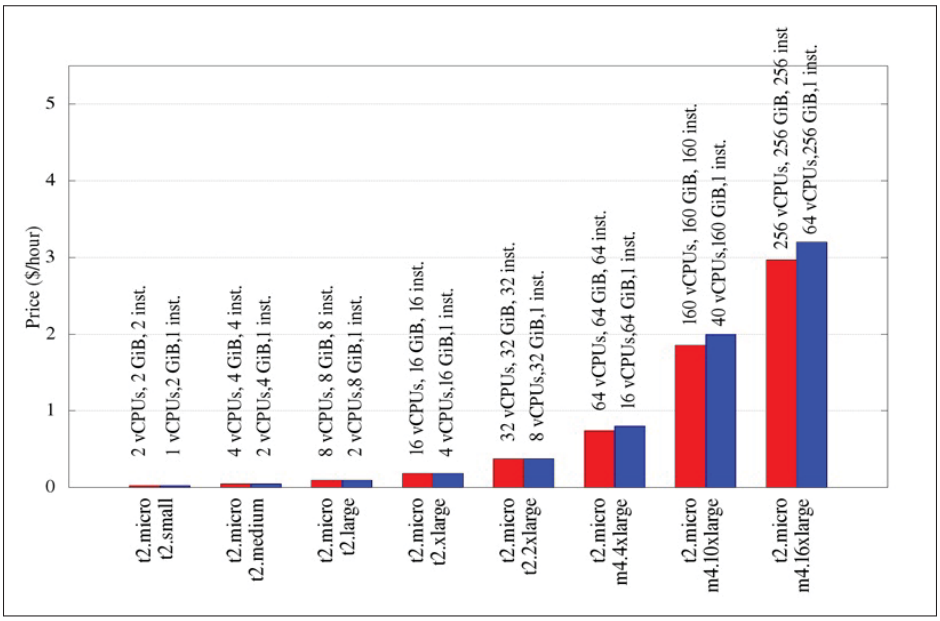


Figure 3.6 Prix/taille des instances et comparaison avec les instances t2.tiny (USA Ouest, Oregon)
Tiré de Ghrada *et al.* (2018)

• **Résultats de l'étude :** Nous pouvons résumer les résultats de l'étude sur les instances de Amazon EC2 comme suit :

- Les coûts d'instance varient considérablement d'un emplacement à l'autre.

- La pile logicielle a un impact important sur le coût de l'instance.
- La capacité de traitement d'un VNF n'est pas nécessairement proportionnelle à la quantité de ressources.
- La capacité de traitement d'un VNF varie considérablement d'une fonction à l'autre.
- Les petites instances (en termes de taille) sont plus rentables, par conséquent, s'il n'y a pas de coût de synchronisation, le déploiement d'instances multiples est plus rentable et offre une capacité de traitement plus élevée.

Compte tenu des résultats ci-dessus, il est important de développer des solutions d'approvisionnement de SFCs capables de trouver le meilleur compromis entre le coût (comprenant le prix de l'instance, la synchronisation et les coûts de bande passante) et la capacité de traitement. Dans ce qui suit, nous proposons un programme linéaire entier pour résoudre la phase de mappage de SFCs et ensuite deux solutions heuristiques pour traiter le problème pour les cas à grande échelle.

3.4 Phase de mappage : formulation du problème

Dans cette section, nous formulons le problème de mappage de SFCs comme un programme linéaire entier (*Integer Linear Program* - ILP) dont l'objectif est de maximiser les bénéfices du fournisseur de SFCs qui sont égaux à ses revenus moins les coûts opérationnels à savoir les coûts de déploiement d'instances et de la bande passante. Cette formulation étend celle proposée par Ghrada (2018) en considérant les bénéfices du fournisseur de SFCs et non seulement les coûts opérationnels. Nous utilisons ainsi la même notation adoptée par Ghrada (2018) et leurs formules pour les coûts opérationnels ainsi que les contraintes de mappage.

L'infrastructure physique détenue par le fournisseur de SFCs est constituée de plusieurs POPs répartis géographiquement. L'infrastructure est modélisée par un graphe $G = (N, P)$ où $N = \{0, 1, ..., |N|\}$ représente l'ensemble des POPs et $P = \{(m, n) \in (N \times N) \mid m$ et n sont directement connectés$\}$ désigne l'ensemble des liens physiques qui relient les POPs. Chaque POP $n \in N$ contient une quantité de ressources physiques C_n exprimée comme le nombre maximal d'ins-

Tableau 3.1 Table de notation

Symbol	Definition
$G = (N, P)$	Graphe G où N est un ensemble de nœuds et P est un ensemble de liens physiques
$V = (I, L)$	Le graphique du réseau virtuel V avec I l'ensemble des instances VNF and L est l'ensemble des liens virtuels
C_n	Capacité disponible au POP $n \in N$ exprimée en nombre d'instances
B_{mn}	Capacité de bande passante du lien physique reliant nœuds m et n
$b_{i,j}$	Besoin en bande passante du lien virtuel connectant les instances i et j
δ_{im}	Coûts de déploiement par unité de temps pour l'instance VNF i dans le POP m
$\Delta_{m,n}$	Coût de bande passante par unité de bande passante du lien physique (m, n)
δ'_{im}	prix par unité de temps pour l'instance VNF i dans le POP m
$\Delta'_{i,j}$	prix de bande passante par unité de bande passante du lien virtuel (i, j)
f_{im}	Constante booléenne définie sur 1 si l'instance VNF i doit être intégrée dans le nœud m
x_{im}	Variable de décision booléenne indiquant si l'instance i est intégrée dans le nœud m
$y_{ij,mn}$	variable de décision booléenne indiquant si le lien virtuel (i, j) est mappé dans le lien physique (m, n)
$\mathbb{R}$	Revenu
$\mathbb{C}$	Coût opérationnel

tances de type t2.tiny que le POP peut héberger. Une instance t2.tiny contient respectivement 1 vCPU, 1 GiB de mémoire et 1 GB de disque. Un lien physique $(m, n) \in P$ qui relie le POP m au POP n a une bande passante B_{mn}. La topologie virtuelle est représentée par un graphe $V = (I, L)$ où $I = \{0, 1, ..., |I|\}$ est l'ensemble des instances virtuelles dans la chaîne et L est l'ensemble des liens virtuels qui les relient.

Chaque instance VNF $i \in I$ a un besoin en ressources de 1 vCPU, 1 GiB de mémoire et 1 GB de stockage. Chaque lien virtuel $(i, j) \in L$ a une exigence de bande passante b_{ij}.

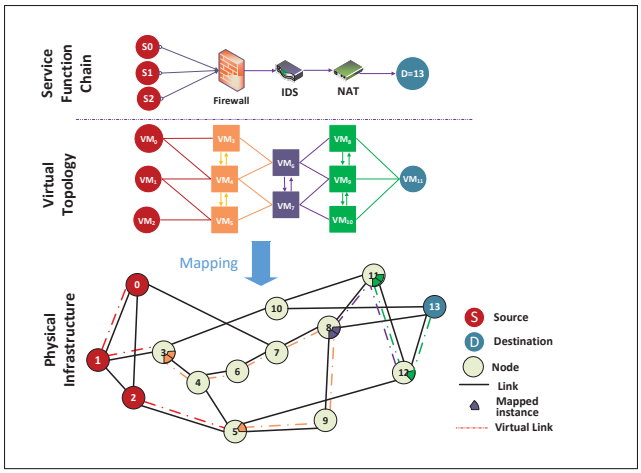


Figure 3.7 Problème de mappage de SFC

On définit la variable de décision $x_{im} \in \{0, 1\}$ qui précise si l'instance VNF i est placée dans le POP m. On définit aussi la variable de décision $y_{ij,mn} \in \{0, 1\}$ qui est égale à 1 si le lien virtuel (i, j) utilise le lien physique (m, n).

Il faut d'abord respecter l'ensemble de contraintes suivantes :

• **Contrainte de mappage des points de terminaison du SFC :** Les points de terminaison du SFC (les sources et les destinations) doivent être intégrés dans des POP spécifiques indiqués dans la requête du SFC. Nous définissons ainsi la variable booléenne f_{im} (fournie comme entrée de l'ILP) qui est égale à 1 si l'instance i est un point de terminaison qui doit être hébergé dans le POP m. L'équation suivante utilise cette contrainte :

$$x_{im} \geq f_{im} \quad \forall m \in N, \ \forall i \in I \tag{3.1}$$

• **Contrainte de mappage d'instance :** Cette contrainte garantit que chaque instance VNF i est intégrée une et une seule fois. Il peut être exprimé comme :

$$\sum_{m \in N} x_{im} = 1 \qquad \forall i \in I \tag{3.2}$$

• **Contrainte de capacité des ressources :** cette contrainte garantit que tout POP d'hébergement dispose de suffisamment de ressources pour héberger les instances des VNFs.

$$\sum_{i \in I} x_{im} \leq C_m \qquad \forall m \in N \tag{3.3}$$

où C_m représente la capacité disponible au POP m.

• **Contrainte de bande passante :** il faut aussi s'assurer que la bande passante nécessaire pour embarquer tous les liens virtuels dans un lien physique ne dépasse pas sa bande passante. Cette contrainte est exprimée comme suit :

$$\sum_{i,j \in L} y_{ij,mn}\, b_{ij} \leq B_{mn} \quad \forall\, (m,n) \in P \tag{3.4}$$

• **Contrainte de conservation de flux :** nous devons également nous assurer que le trafic entrant vers un nœud physique est égal à son trafic sortant à moins que ce POP ne soit une source ou une destination. Cette contrainte peut être exprimée par :

$$\begin{aligned} &\sum_{(n,m) \in P} \sum_{(i,j) \in L} y_{ij,nm}\, b_{ij} - \sum_{(i,j) \in L} x_{jm}\, b_{ij} \\ &= \sum_{(m,n) \in P} \sum_{(i,j) \in L} y_{ij,mn}\, b_{ij} - \sum_{(i,j) \in L} x_{im}\, b_{ij} \quad \forall\, m \in N \end{aligned} \tag{3.5}$$

• **Fonction objectif :** La fonction objectif vise à minimiser les bénéfices du fournisseur de SFCs. Elle est égale au revenu $\mathbb{R}$ moins les coûts opérationnels $\mathbb{C}$ du SFC. Elle peut être exprimée comme suit :

$$J = \max_{\substack{(x_{im})_{i \in I, m \in N} \\ (y_{ij,mn})_{(i,j) \in L, (m,n) \in P}}} (\mathbb{R} - \mathbb{C}) \tag{3.6}$$

Dans ce qui suit, nous fournissons plus de détails sur la façon de calculer le revenu et les coûts opérationnels. Le fournisseur de SFCs collecte un revenu pour chaque SFC qui dépend principalement de la quantité de ressources allouées à la SFC. Ce revenu peut être exprimé

comme suit :

$$\mathbb{R} = \sum_{m \in M} \sum_{i \in I} x_{im}\ \delta'_{im}\ +\ \sum_{(i,j) \in L} \sum_{(m,n) \in P} b_{ij}\ \Delta'_{i,j} \tag{3.7}$$

où δ'_{im} est le prix de vente (exprimé en dollars par unité de temps) de l'instance i dans le POP m. $\Delta'_{i,j}$ désigne le prix de vente en dollars par unité de bande passante et unité de temps pour le lien virtuel (i, j). Par ailleurs, le coût opérationnel des instances et des liens est le coût d'exécution des instances sur l'infrastructure et de la bande passante consommée par les liens virtuels qui les connectent. Il peut être exprimé ainsi :

$$\mathbb{C} = \sum_{m \in M} \sum_{i \in I} x_{im}\ \delta_{im}\ +\ \sum_{(i,j) \in L} \sum_{(m,n) \in P} y_{ij,mn}\ b_{ij}\ \Delta_{m,n} \tag{3.8}$$

où δ_{im} est le coût de déploiement (exprimé en dollars par unité de temps) de l'instance i dans le POP m. Le premier terme de l'équation (Eq. 3.8) représente le coût total de déploiement des instances VNF. Le deuxième terme est le coût de la bande passante consommée par les liens virtuels. $\Delta_{m,n}$ désigne le coût en dollars (par unité de bande passante et unité de temps) pour le lien physique (m, n).

3.5 Phase de mappage - Solutions proposées

Dans cette section, nous abordons ce problème *NP*-difficile en proposant deux solutions heuristiques : un algorithme de base (appelé 'Baseline') et un algorithme plus sophistiqué d'approvisionnement basé sur la composition du SFC (appelé 'SPIN'). Les deux solutions supposent l'existence de plusieurs sources dans le SFC et une seule destination afin de simplifier le problème. Elles visent à minimiser les coûts opérationnels du fournisseur de SFCs tout en garantissant que les demandes de SFCs acceptées respectent l'exigence de délai de bout en bout et de bande passante. Dans ce qui suit, nous fournissons plus de détails sur les deux algorithmes.

3.5.1 Solution 1 : Algorithme de base (Baseline)

L'algorithme de base (Baseline) est un algorithme intuitif qui vise à satisfaire les exigences du SFC en termes de ressources (par exemple, CPU, mémoire, bande passante) et de délai de bout en bout et bande passante tout en minimisant les coûts des instances.

L'algorithme procède avec les étapes suivantes. La première étape consiste à estimer le nombre d'instances et de liens virtuels requis pour l'ensemble de la chaîne. Le nombre d'instances est simplement égal au nombre d'instances t2.micro nécessaires pour traiter le débit de paquets arrivant. La capacité de traitement d'une instance t2.micro est estimée à l'aide de la technique décrite dans la section 3.3. Une fois le nombre d'instances pour chaque VNF est estimé, la topologie virtuelle est construite.

La deuxième étape consiste à allouer des ressources pour cette topologie virtuelle, comme indiqué dans Algorithm 3.1. Pour chaque instance source de la topologie virtuelle, nous commençons par intégrer les nœuds virtuels (c'est-à-dire instances) qui lui sont connectés (c'est-à-dire voisins). Pour chacune de ces instances, nous intégrons récursivement ses voisins en appelant récursivement la fonction $EmbedNeighbors(instance\ i)$ (Algorithm 3.2).

La complexité de cet algorithme récursif est de $O(|I|^2)$ où $|I|$ est le nombre d'instances virtuelles dans la topologie virtuelle.

Algorithme 3.1 Algorithme de base (Baseline)

Input : Virtual topology V= (I,L)
Input : Placement constraint $(f_{im})_{i\in I, m\in N}$
Input : Virtual Topology Destination $d \in N$
Output : Boolean Embedded
1 **foreach** $i \in I$ *such that i is a source (i.e.,* $\sum_{m\in N} f_{im} = 1$*)* **do**
2 $s \Leftarrow$ the hosting physical node of source ;
3 instance i (i.e.,$f_{is} = 1$) ;
4 $x_{is} \Leftarrow 1$;
5 **return** EmbedNeighbors(i) ;
6 **end foreach**

Algorithme 3.2 EmbedNeighbors(instance i)

```
1  s ⇐ Physical node hosting instance i;
2  foreach j ∈ neighbors(i) (Embedding instances connected to i) do
3      if j is not embedded then
4          Find m such that m ∈ ShortestPath(s, d) & C_m ≥ 1 &
            PathBandwidth(s, m) ≥ b_{i,j}; if m exists then
5              if m exists then
6                  x_{jm} ⇐ 1 (Embed j in m); y_{ij,sm} ⇐ 1 (Embed virtual link (i, j) in
                    physical path (s, m)); C_m ⇐ C_m − 1 (Update the node capacity);
7              else
8                  return False (Instance j is not embeddable);
9              end if
10         end if
11     end if
12 end foreach
13 foreach j ∈ neighbors(s) do
14     if j is not embedded then
15         return EmbedNeighbors(j) (Embedding instances connected to j);
16     end if
17 end foreach
18 return True (all instances were embedded);
```

3.5.2 Solution 2 : Algorithme d'approvisionnement basé sur la décompoSition des SFCs (SPIN)

Cet algorithme est appelé *SFC decompoSition-based Provisioning* (SPIN) et se déroule en quatre phases (Algorithm 3.3). Dans la première phase, nous estimons le nombre d'instances pour chaque VNF et estimons le nombre de liens virtuels comme le fait l'algorithme de base.

La deuxième phase est la phase de décomposition où la topologie virtuelle est divisée en sous-chaînes où chaque sous-chaîne est une chaîne d'instances VNF qui contient une seule instance de chaque type VNF et connecte une source à une destination.

La troisième phase est la phase de mappage de sous-chaîne (Algorithm 3.4) où chaque sous-chaîne est intégrée dans le chemin le plus court entre la source et la destination de la sous-chaîne notée P. Le chemin P est celui sélectionné comme celui avec le coût le plus bas et qui a un

délai satisfaisant l'exigence de délai e2e de la chaîne et a suffisamment de ressources pour intégrer la sous-chaîne ($FreeInst(P)$ est le nombre d'instances libres dans le chemin P et $NumberInstances(SC_k)$ est le nombre d'instances nécessaires à la sous-chaîne SC_k). Les liens virtuels destinés à acheminer le trafic de synchronisation sont alors provisionnés entre les instances VNF de même type (Function $EmbedSynchronizationVirtualLinks(V)$).

La dernière phase est la phase d'optimisation 3.5 qui consiste à sélectionner chaque instance et à explorer la possibilité de la migrer dans l'un des nœuds physiques voisins de son emplacement physique actuel. L'objectif est de réduire davantage les coûts d'exploitation et de synchronisation (Eq. 3.6) tout en garantissant toujours que la bande passante demandée et le délai e2e sont satisfaits.

La complexité de l'algorithme SPIN est de $O(K)$ où K est le nombre de sous-chaînes. La complexité de la phase d'optimisation est de $O(|V|)$ où V est le nombre d'instances virtuelles dans la topologie virtuelle.

Algorithme 3.3 SPIN

```
   Input  : Virtual topology V = (I, L)
   Input  : Placement constraint (f_im)_{i∈I,m∈N}
   Input  : Virtual Topology Destination d ∈ N
   Output : Boolean Embedded, VLEmbedded
 1 Decompose V into K subchains (SC_k)_{(k=1..K)}
 2 repeat
 3 |   Embedded ⇐ EmbedSubchain(SC_k);
 4 |   k ⇐ k + 1;
 5 until Embedded = False || k = K + 1;
 6 VLEmbedded ⇐ EmbedSynchronizationVirtualLinks(V);
 7 if Embedded & VLEmbedded)=True (Embedding is succesful) then
 8 |   if Embedded & VLEmbedded)=True (Embedding is succesful) then
 9 |   |   Optimization(V) (optimization phase);
10 |   |   return True;
11 |   else
12 |   |   return False;
13 |   end if
14 end if
```

Algorithme 3.4 EmbedSubchain(subchain SC_k)

```
P ⇐ Find path with minimal cost such that delay(P) ≤ delay(SC_k &
  FreeInst(P) ≤ NumberInstances(SC_k) & Bandwidth(P) ≥ Bandwidth(SC_k);
if P exists then
    Embed SC_k in P;
    return True (SC_k is successfully embedded);
else
    return False (SC_k is not embeddable);
end if
```

Algorithme 3.5 Optimization (VirtualTopology V)

```
foreach i ∈ V (Parse all instances) do
    n ⇐ Physical node hosting i;
    foreach m ∈ neighbors(n) (Explore migrating i to neighboring nodes) do
        Cost ⇐ CurrentVCost(V) (Compute Embedding Cost Eq. 3.6);
        NewCost ⇐ VCost(V, i, m) (Compute cost assuming i is hosted in m);
        if CheckConstraints(V) & NewCost < Cost (all resource constraints
          should be satisfied) then
            Migrate(i, m) (migrate instance i to physical node m;;
        end if
    end foreach
end foreach
```

3.6 Évaluation de performances

3.6.1 Mise en place de la simulation

Afin d'évaluer les performances des algorithmes proposés, nous avons développé un simulateur qui repose sur le langage C afin de simuler la topologie physique et réaliser la traduction et le mappage des requêtes SFC. Chaque simulation suppose l'arrivée des demandes pendant deux mois.

L'infrastructure physique est censée contenir 25 nœuds, chaque nœud ayant une capacité d'hébergement définit aléatoirement entre 50 et 100 instances de type 't2.micro'. Les nœuds

sont connectés avec des liens de 10 Gbps avec des délais de propagation fixés aléatoirement entre 10 et 50 ms.

Les SFCs ont été générés aléatoirement avec un taux d'arrivée moyen compris entre 0,1 et 0,15 rps suivant une distribution de Poisson. La durée de vie moyenne des requêtes suit une distribution exponentielle d'une heure en moyenne. Le nombre moyen de VNFs par SFC est de 10 avec un nombre moyen de sources d'environ 7. La demande en termes d'arrivée de paquets pour chaque SFC est générée aléatoirement entre 2000 et 120 000 paquets par seconde.

L'exigence de délai de bout en bout d'une requête SFC est calculée comme suit $\max_{s,d}(t_{s,d}) \times 130\%$ où $t_{s,d}$ est la latence du chemin entre une source s et une destination d où s et d sont une source et une destination de la requête SFC. Cela garantit que, théoriquement, l'exigence de délai de bout en bout entre les sources du SFC et sa destination pourrait être satisfaite tant qu'il est 30% plus élevé que tout chemin entre les sources et les destinations de la demande.

Nous supposons également que nous avons 9 types de VNFs. La capacité de traitement des paquets de chaque type de VNF est générée aléatoirement entre 2000 et 12000 paquets par seconde (pps) lors de l'exécution sur une instance de type 't2.micro'. Le coût de synchronisation entre les instances de même type est de 0.01$/heure multiplié par le type de l'instance. Nous avons utilisé les prix des instances Amazon EC2 comme coûts d'instance. Le revenu de l'instance comme le coût de l'instance plus 0.1$/heure. cela signifie qu'il y a 0,1$/heure de profit pour le fournisseur de SFC pour chaque instance.

Dans ce qui suit, nous présentons les résultats générés pour les deux algorithmes proposés dans le cadre de la configuration de simulation décrite précédemment.

3.6.2 Résultats des simulations

Nous comparons d'abord les performances des algorithmes Baseline et SPIN pour un taux d'arrivée de 0,03 requêtes par seconde (rps).

Comme le montrent les figures 3.8 et 3.9, SPIN accepte 25% de requêtes en plus et conduit aussi à une utilisation plus élevée du CPU d'environ 37% dans toute l'infrastructure.

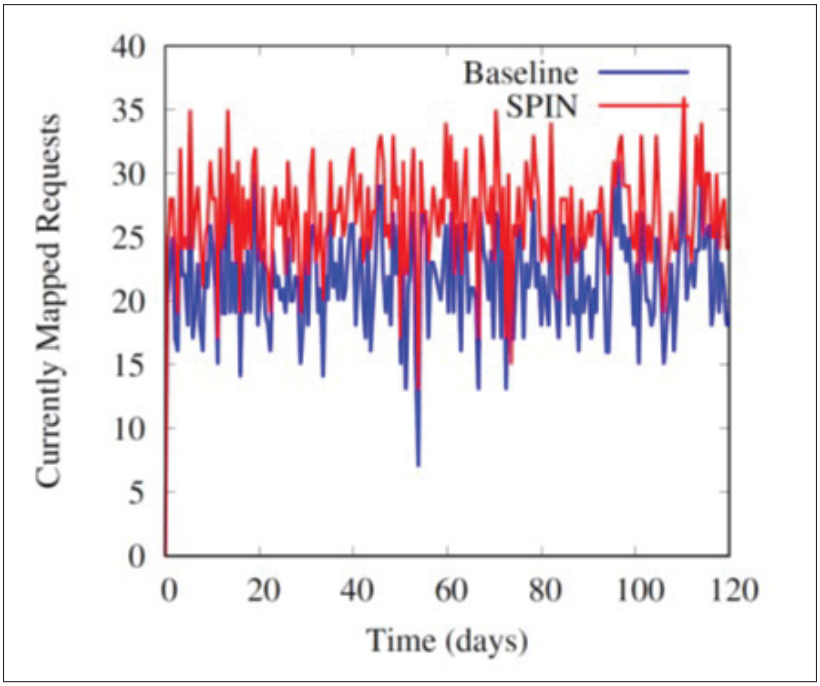


Figure 3.8 Nombre de demandes mappées au fil du temps (taux d'arrivée des demandes : 0,03 rps)

Pour évaluer davantage les performances des deux algorithmes heuristiques proposés considérant différents scénarios proposés, nous avons calculé les métriques suivantes en fonction du taux d'arrivée des demandes de SFCs :

- Ratio d'acceptation : ce ratio est calculé comme étant le rapport du nombre de SFCs acceptés par rapport au nombre total de requêtes SFC reçues. Les requêtes acceptées font référence à celles pour lesquelles l'algorithme a réussi à trouver suffisamment de ressources pour le SFC tout en satisfaisant ses exigences de délai de bout en bout et bande passante.
- Utilisation de l'infrastructure : c'est le pourcentage de ressources CPU utilisée par rapport aux ressources totales disponibles de CPU.

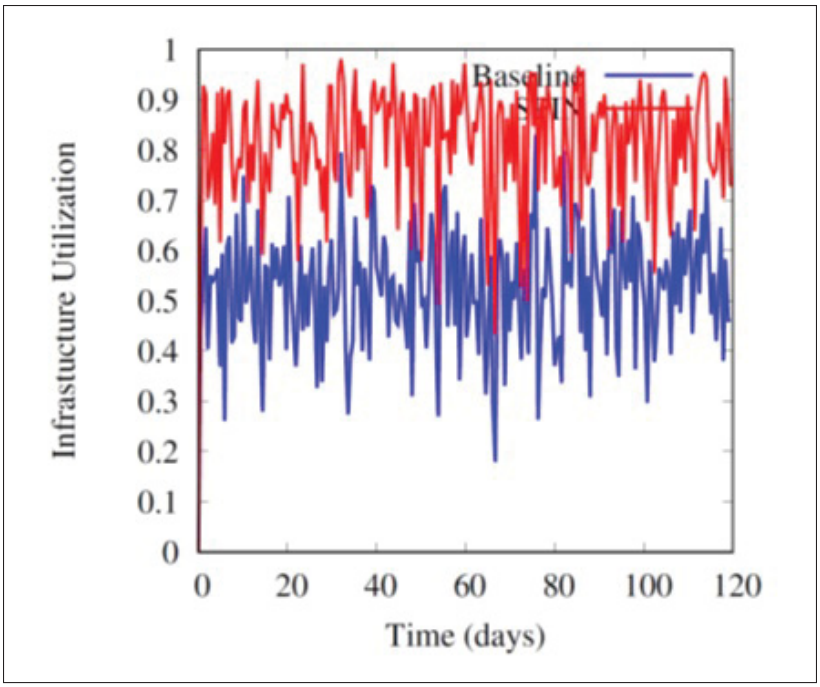


Figure 3.9 Utilisation du processeur de l'infrastructure au fil du temps (taux d'arrivée des demandes : 0,03 rps)

- Bénéfice cumulé : c'est le bénéfice est calculé comme le chiffre d'affaires du fournisseur de SFCs moins les coûts opérationnels, y compris les coûts de l'instance, de la bande passante et de la synchronisation. Le Bénéfice cumulé est calculé pour la durée de l'expérience.
- Délai moyen de bout en bout (e2e) : le délai moyen de bout en bout entre les sources et les destinations des demandes de SFCs qui ont été intégrées avec succès tout au long de l'expérience.

Dans ce qui suit, nous fournissons et discutons les résultats obtenus pour chaque métrique.

La première métrique considérée est le taux d'acceptation qui est représentée dans la Figure 3.10. La figure montre que, même avec un faible taux d'arrivée des demandes, l'algorithme Baseline ne parvient pas à accueillir 50% des demandes alors que l'algorithme SPIN réussit à accueillir jusqu'à 65% des demandes. Cela signifie que même si l'utilisation de l'infrastructure est faible

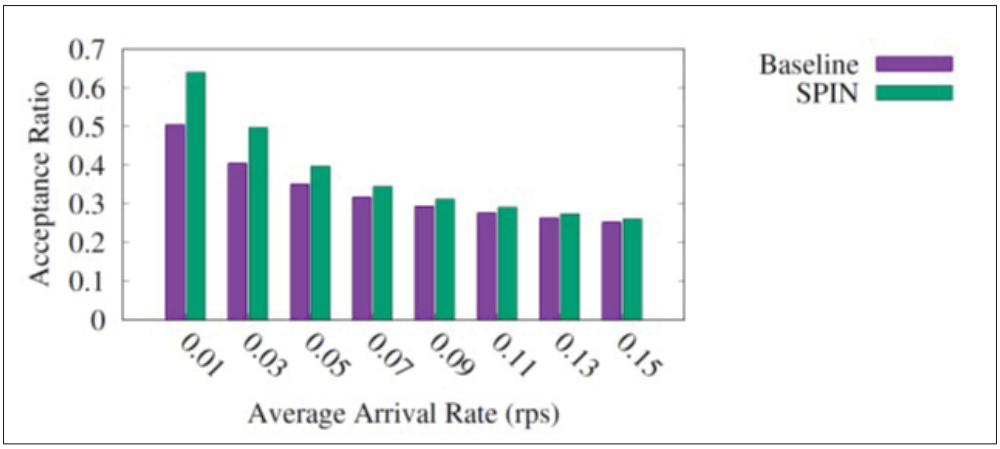


Figure 3.10 Ratio d'acceptation

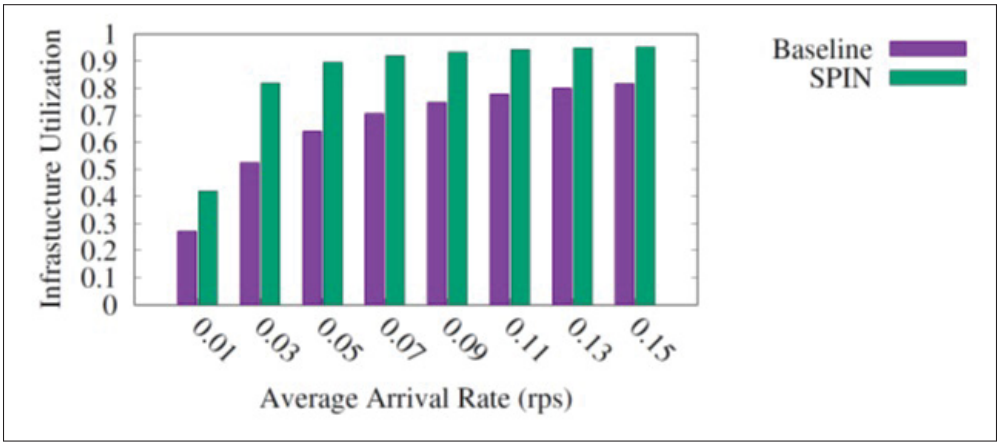


Figure 3.11 Utilisation de l'infrastructure

et que les ressources sont disponibles (Figure 3.11), l'algorithme Baseline, contrairement à l'algorithme SPIN, n'est pas en mesure d'exploiter efficacement les ressources disponibles.

Au fur et à mesure que le taux d'arrivée augmente, le taux d'acceptation diminue pour les deux algorithmes à mesure que l'infrastructure devient saturée comme le montre la Figure 3.11. Cependant, SPIN surpasse toujours le Baseline en termes de taux d'acceptation.

De plus, comme l'illustre la Figure 3.11, SPIN accepte jusqu'à 25% de demandes en plus pour de faibles taux d'arrivée, ce qui montre qu'il permet d'exploiter efficacement les ressources de

l'infrastructure par rapport à Baseline. Il est à noter que pour des taux d'arrivée élevés, SPIN réussit à atteindre 95% d'utilisation par rapport à 80% d'utilisation pour Baseline.

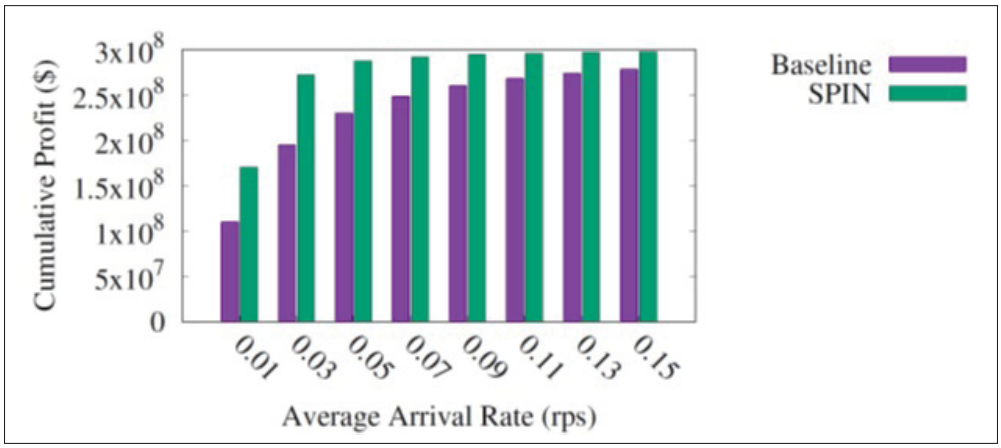


Figure 3.12 Bénéfice cumulé

Nous étudions également le bénéfice cumulé généré par chacun des deux algorithmes (Figure 3.12). La figure montre que le profit généré par SPIN dépasse jusqu'à 30% celui généré par le Baseline, en particulier pour les faibles taux d'arrivée.

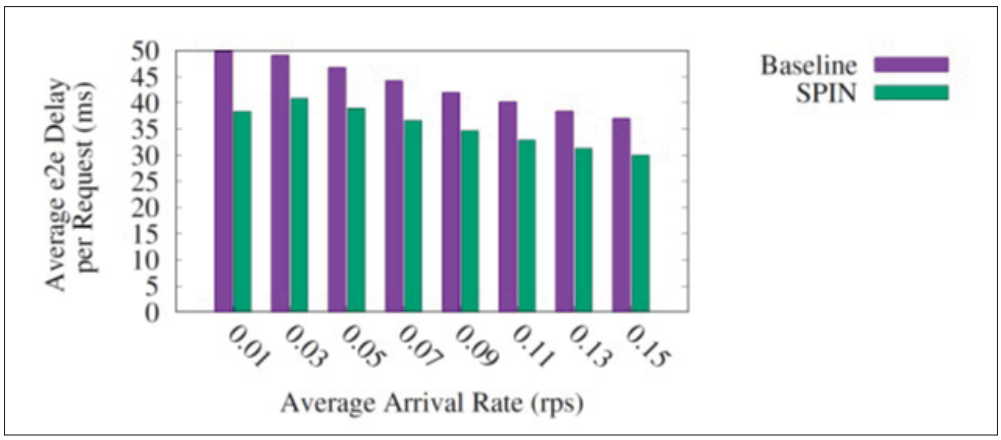


Figure 3.13 Délai moyen de bout en bout par demande

Enfin, la Figure 3.13 montre le délai moyen de bout en bout par requête pour les SFCs acceptés. Cela montre qu'en cas de faible utilisation, SPIN réduit jusqu'à 35% le délai de bout en bout et jusqu'à 25% pour des taux d'arrivée élevés. Cela montre que SPIN ne satisfait pas seulement les

exigences des demandes en termes de délai de bout en bout, mais le réduit encore davantage par rapport à Baseline.

3.7 Conclusion

La sélection du bon emplacement pour les VNFs ainsi que le nombre et le type d'instances est un défi majeur pour les fournisseurs de services infonuagiques puisque cela a un impact important non seulement sur la performance, mais également sur le coût. Dans ce chapitre, nous avons étudié ces compromis à l'aide des instances Amazon EC2 à usage général. Par exemple, nous avons remarqué que les micro-instances (petites instances) sont plus rentables. Nous constatons également que les performances d'une machine virtuelle ne sont pas toujours proportionnelles à la quantité de ressources allouées. Par conséquent, nous avons constaté que l'approvisionnement de plusieurs petites instances, lorsque la fonction pourrait être distribuée, fournirait de meilleures performances que les grandes instances et aussi avec un coût nettement inférieur.

En outre, nous avons étudié l'allocation des ressources axée sur le profit en modélisant mathématiquement le problème comme un programme linéaire entier et en proposant deux heuristiques Baseline et SPIN. SPIN est un algorithme plus sophistiqué qui permet d'améliorer les performances du mappage avec plus de chaînes acceptées et donc plus de profits.

CHAPITRE 4

ROUTAGE MULTI-DOMAINES COLLABORATIF DANS UN ENVIRONNEMENT SDN

4.1 Introduction

Plusieurs applications futures (par exemple, la télé-chirurgie, la vidéoconférence, la télédétection par satellite, les véhicules autonomes) nécessitent des garanties strictes sur la bande passante, le délai de bout en bout et le coût monétaire cautionné. Dans ce contexte, un enjeu majeur du côté des fournisseurs d'Internet est de fournir des garanties strictes en termes de performance afin de satisfaire les exigences des applications tout en optimisant l'utilisation du réseau et en maximisant les profits (Famaey, Latré, Wauters & De Turck, 2014; Karakus & Durresi, 2017). Malheureusement, même avec l'émergence et l'adoption croissante de nouvelles technologies telles que la réseautique définie par logiciel (SDN) (Nandan, Porowski, Liu & Forsman, 2018), l'internet ne parvient pas à garantir des performances de bout en bout, principalement, parce qu'il est fragmenté en plusieurs domaines administratifs qui sont gérés de manière indépendante.

Dans un domaine administratif unique, le contrôleur SDN a une vue globale du réseau et pourrait facilement ajuster les décisions de routage dynamiquement et à la demande et ainsi satisfaire les objectifs de performance requis. Cependant, dans un scénario multi-domaine, les domaines sont gérés séparément par différents opérateurs de réseau. Chaque domaine a un contrôleur qui est responsable du routage des paquets dans son domaine indépendamment des autres. En conséquence, il ne prend pas en considération :

- les décisions de routage dans d'autres domaines,
- l'état actuel des autres domaines en termes d'utilisation et de congestion,
- les exigences globales de performance de l'application (par ex., la bande passante et le délai de bout en bout associés aux applications),
- et les coûts monétaires, c'est-à-dire, le coût encouru lorsqu'un flux utilise un ou plusieurs domaines.

Par conséquent, il n'est pas possible de garantir les performances des applications réseau lorsque leurs flux de trafic associés traversent différents domaines administratifs.

Une solution prometteuse pour résoudre ce problème consiste à proposer des techniques de routage collaboratif efficaces entre différents domaines (Moufakir, Zhani, Gherbi & Bouachir, 2022). La technologie SDN offre la possibilité de publier facilement des informations à propos de chaque domaine via des interfaces de programmation d'application (*eastbound APIs*) des contrôleurs SDN. Les contrôleurs des domaines concernés peuvent publier des informations pour collaborer afin de prendre les meilleures décisions et pour améliorer la performance globale de l'application. Bien entendu, les informations échangées ne devraient pas être exhaustives pour de nombreuses raisons telles que la sécurité et la concurrence sur le marché entre les opérateurs de réseau et pourraient donc être limitées à certaines statistiques ou des informations spécifiques. Ainsi, chaque opérateur doit partager certaines informations sur son domaine reflétant son état actuel.

La plupart des recherches effectuées sur la collaboration entre plusieurs domaines ont soit envisagé une collaboration contrôleur-à-contrôleur dans laquelle les contrôleurs des domaines impliqués échangent directement des données entre eux (Tootoonchian & Ganjali, 2010; Phan *et al.*, 2013; Berde *et al.*, 2014; Medved *et al.*, 2014; Jahan *et al.*, 2018; Yin *et al.*, 2012; Marconett & Yoo, 2015), soit une collaboration basée sur un courtier (*broker*) où le courtier collecte les données et prend les décisions de routage entre les différents domaines (Sherwood, Gibb, Yap, Appenzeller, Casado, McKeown & Parulkar, 2009; Vilalta *et al.*, 2016; Kotronis *et al.*, 2015; Gerola *et al.*, 2016).

Toutes ces propositions visent principalement à assurer le routage entre plusieurs domaines sans apporter de garanties sur les performances de bout en bout. Ils ne prennent pas en considération les coûts de transmission des flux sur plusieurs domaines et ne tentent pas de l'optimiser.

Pour remédier à ces limitations, la principale contribution de ce chapitre est la conception et la mise en place d'un nouveau mécanisme de routage collaboratif multi-domaine qui exploite la collaboration entre les contrôleurs SDN de plusieurs domaines via un courtier chargé de

router les flux sur plusieurs domaines. L'objectif du courtier est de maximiser le nombre de flux acheminés répondant à leurs exigences de performance de bout en bout et en garantissant un coût de transmission de flux minimal.

Dans ce chapitre, nous allons présenter une description détaillée du problème et mettre en évidence les défis associés, suivi de la nouveauté de ce travail. Nous décrivons par la suite la conception et la mise en œuvre de la solution, suivi des résultats expérimentaux et des conclusions.

4.2 Description du problème

Cette section aborde plus en détail le problème traité et met en évidence les différents défis. Dans les réseaux à grande échelle, les paquets de données peuvent traverser plusieurs domaines SDN gérés par différents opérateurs de réseau. Cependant, sans collaboration entre les opérateurs et sans possibilité de partager des informations internes sur les caractéristiques et l'état de leurs réseaux, chaque domaine est vu comme une "boîte noire".

Le manque de collaboration entraîne de nombreuses limitations :

- le manque de connaissance de la topologie pour déterminer un chemin optimal de bout en bout,
- le manque d'informations sur l'état des chemins dans les différents domaines (par exemple, l'utilisation, retard, bande passante),
- l'incapacité de connaître à l'avance les chemins possibles pour atteindre la destination sous chaque domaine,
- et l'incapacité à garantir des performances optimales pour les applications pour lesquelles les paquets doivent traverser plusieurs domaines administratifs.

La figure 4.1 décrit un scénario simple de plusieurs domaines administratifs (D_1 à D_5). Nous supposons qu'il existe un flux provenant d'une "source" dans le domaine D_1 vers une "destination" dans le domaine D_5. Sans collaboration, l'état de chaque domaine est masqué de tous les autres domaines. Comme le montre la figure, il existe plusieurs chemins inter-domaines (c'est-à-dire,

chemin 1, chemin 2, chemin 3) pour atteindre D_5 à partir de D_1. Le contrôleur de D_1 n'a qu'une vue complète de sa propre topologie de domaine et n'a connaissance d'aucune information ou statistique sur le domaine suivant (par exemple, utilisation, délai). Par conséquent, les décisions d'acheminement prises peuvent ne sont pas nécessairement optimales et peuvent entraîner un délai de bout en bout plus élevé comme elles ne tiennent compte que des informations internes du domaine D_1 et ignorent l'état des autres domaines.

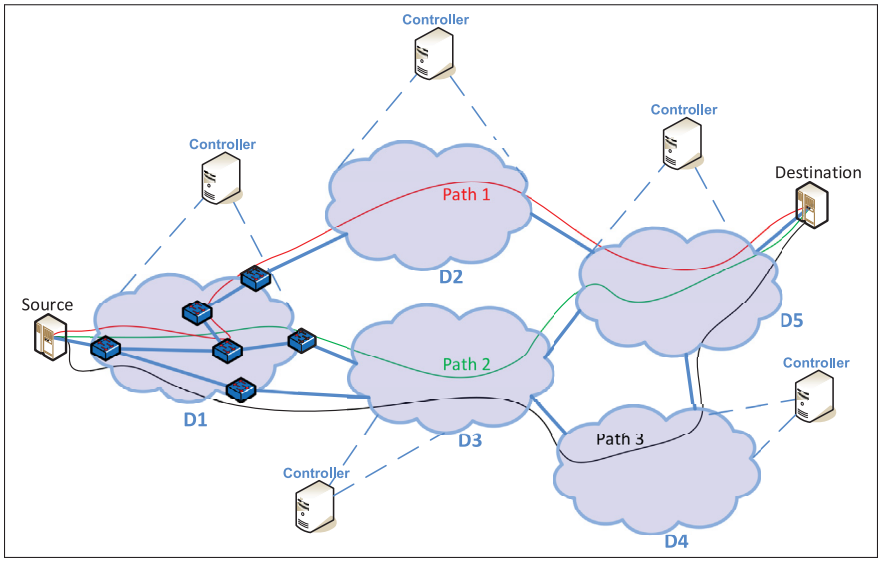


Figure 4.1 Routage de flux à travers plusieurs domaines

Le partage d'informations internes sur les caractéristiques et l'état entre différents domaines permettra de prendre les meilleures décisions de routage. Une telle collaboration peut augmenter considérablement les performances du routage multi-domaine (par ex., réduire la perte de trafic, le délai de bout en bout et maximiser l'utilisation des liens réseau), offrir une meilleure utilisation du réseau et réduire le risque de congestion.

Dans ce chapitre, nous supposons que chaque domaine publie ses informations internes et leurs statuts via un courtier (par exemple, utilisation, délai, bande passante). Ce routage basé sur les courtiers permet de profiter des avantages de la collaboration sans risques majeurs de sécurité.

4.3 Solution proposée et formulation mathématique

Cette section présente l'architecture système proposée pour gérer le routage de flux à travers plusieurs domaines.

L'objectif de l'architecture proposée est de permettre une collaboration efficace entre plusieurs contrôleurs SDN gérant différents domaines administratifs. Comme le montre la Figure 4.2, l'architecture proposée repose sur un courtier qui reçoit régulièrement des mises à jour de statut des différents domaines, puis construit *une vue multi-domaine abstraite* et l'utilise pour calculer la meilleure route pour les flux inter-domaines. Les avantages de l'utilisation du courtier (*Broker*) sont les suivants :

- Il permet à chaque domaine de prendre la meilleure décision de routage en fonction de l'état des autres domaines afin de maximiser le nombre de flux interdomaines satisfaisant leurs exigences de performances et de coût.
- Il construit une relation gagnant-gagnant entre les opérateurs des domaines concernés en évitant une congestion interne potentielle.
- Il utilise les informations partagées pour construire une vue abstraite multi-domaine et l'exploite pour calculer le chemin de routage inter-domaine pour chaque flux.
- Il pré-configure les domaines impliqués dans le routage d'un flux particulier avec le chemin optimal avant l'arrivée du flux à chacun d'eux.

4.3.1 Architecture du système

Comme le montre la Figure 4.2, le courtier proposé est composé de :

- **Module de collecte et de surveillance des chemins** : ce module est responsable de la réception et de la mise à jour des chemins intra-domaines publiés en permanence (liens d'interconnexion entre les nœuds de bordure de chaque domaine), des liens inter-domaines (liens entre différents domaines) et de leur état (par exemple, délai, bande passante). Toutes les informations seront stockées dans une base de données.

- **Path Computation Element** : ce module répond aux requêtes provenant d'un domaine pour calculer un chemin inter-domaine de bout en bout pour un flux particulier afin de satisfaire ses exigences de performance et de coût.

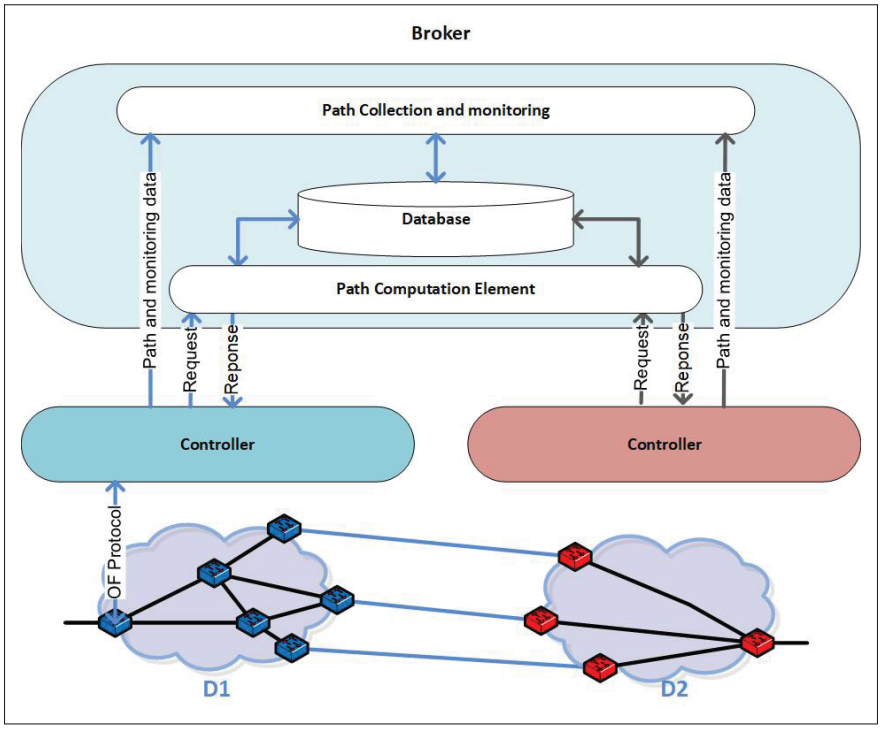
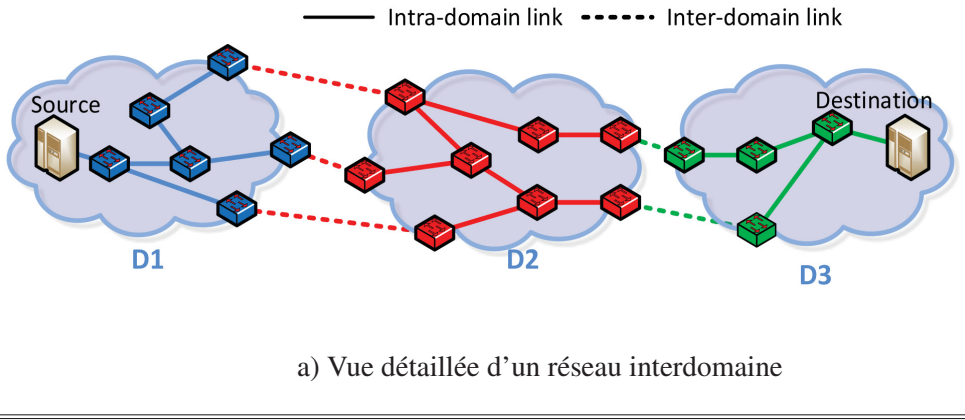


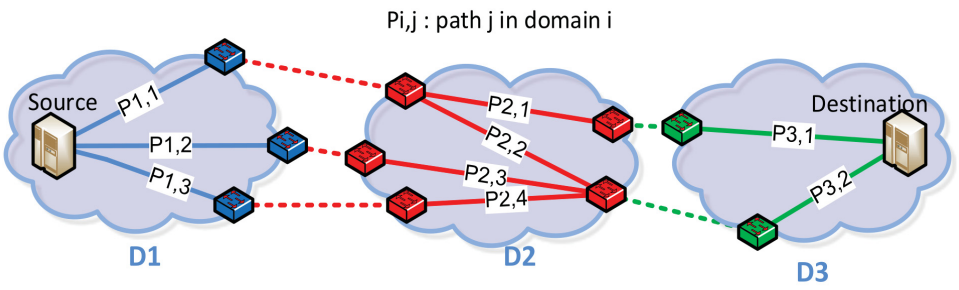
Figure 4.2 Architecture de collaboration entre plusieurs domaines

4.3.2 Les parties prenantes

Comme décrit ci-dessus, nous pouvons identifier deux principaux acteurs qui devraient jouer un rôle afin de mettre en œuvre la solution proposée. Le premier acteur sont les opérateurs de réseaux qui vont coopérer. Le second est le propriétaire du courtier qui pourrait être une alliance d'opérateurs de réseau qui permettrait aux membres de partager certaines informations de domaine reflétant son état actuel afin de prendre les meilleures décisions de routage pour garantir les performances de l'application demandée. Une autorité publique indépendante pourrait être chargée de gérer et de superviser le service de courtage. Si un tel courtier n'est pas mis en œuvre, il ne sera pas possible de garantir des performances de bout en bout et cela empêchera la



a) Vue détaillée d'un réseau interdomaine



b) Vue abstraite d'un réseau multi-domaines

Figure 4.3 Aperçu d'un réseau multi-domaines

mise en œuvre à grande échelle de plusieurs applications futuristes qui nécessitent des garanties strictes sur les performances. Cela est une forte motivation pour les opérateurs de réseaux afin d'envisager le déploiement d'un tel courtier et d'autoriser le partage de certaines informations sur l'état et les performances de leurs réseaux. Nous supposons que des mécanismes de contrôle et de réglementation efficaces sont nécessaires lors de la mise en place du courtier pour assurer une collaboration fluide, juste et équitable entre eux.

4.3.3 Vue abstraite multi-domaine

La Figure 4.3a montre un exemple de vue détaillée de plusieurs domaines réseau avec une source et une destination. La figure montre les nœuds internes et ceux à la bordure, les liaisons

intra-domaines et les liaisons inter-domaines. Cependant, le courtier proposé n'a qu'une vue multi-domaine abstraite comme le montre la Figure 4.3b où chaque domaine est vu comme un ensemble de chemins intra-domaines reliant les nœuds à la bordure ainsi que les liens inter-domaines également. Un chemin intra-domaine $P_{i,j}$ dans le domaine i est identifié par un ID (j) ainsi que par les nœuds à la bordure (source et destination). En conséquence, les détails du chemin, y compris les nœuds et les liens internes, ne sont pas vus par le courtier et ne sont pas partagés par le contrôleur de domaine. La décision de routage sera alors prise en fonction des liaisons inter-domaines et des chemins intra-domaines qui sont partagés par chaque contrôleur.

La vue multi-domaine abstraite est créée selon les étapes suivantes :

- Chaque administrateur de domaine choisit de publier un sous-groupe de ses chemins intra-domaine (une simple représentation des liens intra-domaines interconnectés) comme indiqué dans le tableau 4.1.
- Chaque domaine doit mesurer l'état de tous les chemins publiés et les annoncer par la suite au courtier.
- Les chemins publiés sont déterminés par un identifiant unique sans fournir de détails sur les nœuds internes et les liens qui les composent.
- Avant de transmettre des données via plusieurs domaines administratifs, chaque contrôleur doit demander au courtier de sélectionner les domaines traversés et le chemin complet entre la source et la destination.
- Le courtier informe les contrôleurs impliqués du chemin de routage à utiliser pour le nouveau flux entrant.

4.3.4 Séquence de configuration du flux

La figure 4.4 montre la séquence de configuration des nouveaux flux inter-domaines. Chaque nouveau flux inter-domaine issu d'un commutateur dans un domaine source et ciblant une destination dans un autre déclenche une demande de flux inter-domaine au courtier envoyée par le contrôleur de domaine source. Le courtier calcule le chemin inter-domaines de bout en bout

Tableau 4.1 Informations intra-domaine partagées par chaque domaine

Variable	**Description**
ID du domaine	Identifiant du domaine du commutateur source.
ID du chemin	Identifiant du chemin.
Src. Switch ID	Numéro d'identification de la source (commutateur de bordure) du chemin.
Dest. Switch ID	Numéro d'identification de la destination (commutateur de bordure) du chemin.
Bande passante	Bande passante disponible dans le chemin.
Délai	Délai total du chemin.
Taux de perte	Taux de perte de paquets par nombre total de paquets sur le chemin.
Coût	Coût du chemin.

en fonction des objectifs de performances et du coût requis. Il notifie ensuite les contrôleurs des domaines traversés afin qu'ils installent les règles de transfert dans leurs domaines associés.

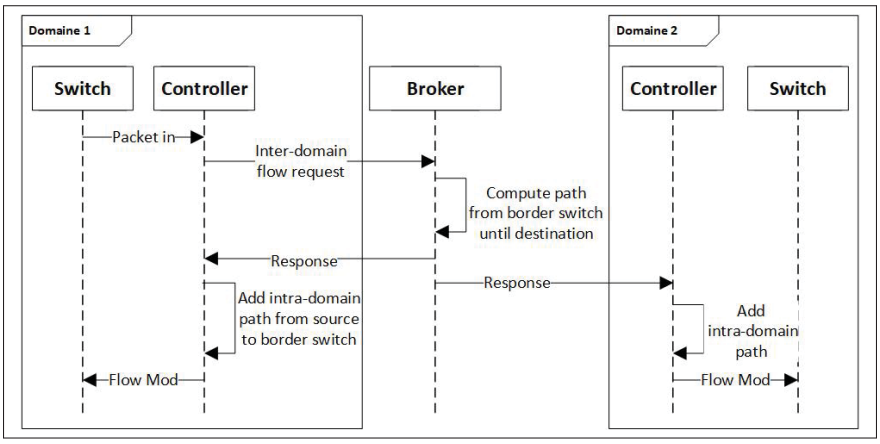


Figure 4.4 Séquence de configuration du flux

4.3.5 Formulation du problème

Dans ce qui suit, nous formulons le problème de routage de flux inter-domaines sous forme d'un programme linéaire en nombres entiers (*Integer Linear Program* - ILP) qui vise à maximiser le nombre de flux planifiés (notés Ω) en garantissant leurs exigences de performance en termes de bande passante, de délai et les exigences en matière de coûts.

Le réseau fédéré constitué de tous les domaines administratifs considérés permettra de créer une vue abstraite du réseau complet. Cette vue abstraite est modélisée comme un graphe non orienté $G(V, A)$, où V est l'ensemble des nœuds représentant les ports des routeurs de bordure et A est l'ensemble des liens représentant les chemins possibles entre les ports des routeurs de bordure.

Considérant que le réseau sera traversé par un ensemble de flux unidirectionnel noté F. Chaque flux $k \in F$ est caractérisé par un ensemble de propriétés comprenant le nœud source s_k, le nœud de destination d_k, la bande passante β_k, le délai maximum de bout en bout δ_k (en ms) et le coût monétaire maximum de routage σ_k. Le tableau 4.2 résume les notations utilisées dans cette formulation.

Tableau 4.2 Table de notation

Variable	**Description**
b_{ij}	available bandwidth between nodes i and j
d_{ij}	delay of the link between nodes i and j
c_{ij}	coût de la liaison entre les nœuds i et j
k	flux $k \in F$
β_k	bande passante du flux k
δ_k	délai maximum demandé pour le flux k
σ_k	coût maximum demandé pour le flux k
Ω	nombre maximum de flux planifiés
x_{ij}^k	$\begin{cases} 1 & \text{si flux } k \text{ utilise le lien (i,j)} \\ 0 & \text{autrement} \end{cases}$
z^k	$\begin{cases} 1 & \text{si le flux } k \text{ est programmé} \\ 0 & \text{autrement} \end{cases}$

Notre objectif est de maximiser Ω, le nombre de flux planifiés avec succès :

$$\Omega = \sum_{k \epsilon F} z^k \tag{4.1}$$

où $z^k \in \{0, 1\}$ est égal à 1 si le flux k est planifié avec succès sur multiples domaines tout en respectant les exigences en termes de performances et de coût.

Pour garantir que les exigences de débit sont satisfaites, les contraintes suivantes doivent être satisfaites :

$$\sum_{i \epsilon A} x_{ij}^k - \sum_{i \epsilon A} x_{ji}^k = \begin{cases} 1 & \text{si } i = s \text{ (source)}, \\ -1 & \text{si } i = d \text{ (sink)}, \\ 0 & \text{autrement} \end{cases} \tag{4.2}$$

$$\sum_{k \epsilon F} x_{ij}^k \beta_k \leq b_{ij} \tag{4.3}$$

$$\sum_{(i,j) \epsilon A} x_{ij}^k d_{ij} \leq \delta_k \tag{4.4}$$

$$\sum_{(i,j) \epsilon A} x_{ij}^k c_{ij} \leq \sigma_k \tag{4.5}$$

$$z^k = \begin{cases} 1 & \text{si } \sum\limits_{(i,j) \epsilon A} x_{ij}^k > 0 \\ 0 & \text{autrement} \end{cases} \tag{4.6}$$

La contrainte (4.2) garantit que les liens entrants et sortants vers un nœud intermédiaire sont égaux.

La contrainte (4.3) garantit que la bande passante utilisée par les flux dans un lien particulier (i, j) ne dépasse pas sa capacité totale de bande passante.

La contrainte (4.4) garantit que le délai total pour un flux k sur tous les liens traversés ne peut pas dépasser le délai maximum requis pour ce flux.

La contrainte (4.5) garantit que le coût de tous les liens traversés ne dépasse pas le coût maximum accepté pour un flux particulier k.

La contrainte (4.6) définit la variable z^k à 1 est le flux k est planifié avec succès, c'est-à-dire qu'au moins x_{ij}^k est égal à 1, ce qui signifie qu'il y a un chemin complet de la source à la destination capable de transmettre le flux k et de satisfaire ses contraintes.

Il est à noter que la maximisation du nombre de flux programmés est un problème NP-difficile. Dans le cas de réseaux à grande échelle et avec un grand nombre de flux inter-domaines, trouver la solution optimale à l'aide de l'ILP peut ne pas être possible. Pour résoudre ce problème, nous proposons dans la section suivante deux algorithmes heuristiques pour la résolution du problème à grande échelle permettant de fournir des solutions quasi-optimales.

4.3.6 Algorithme de routage à vue globale

Dans cette section, nous proposons deux algorithmes heuristiques pour trouver les meilleurs itinéraires des flux à travers plusieurs domaines administratifs de réseau. Le premier algorithme est appelé algorithme de routage à vue globale (*Global View Routing Algorithm* - GlobalRT). Il doit être exécuté par le courtier et exploite la vue abstraite du réseau multi-domaine pour trouver une solution quasi-optimale pour la planification des flux. L'algorithme GlobalRT vise à maximiser le nombre de flux réussis ainsi que satisfaire les performances requises. Il est décrit dans Algorithme 4.1 et comprend les étapes suivantes :

`Étape 1:` lors de la réception d'un nouveau flux k, l'algorithme calcule tous les chemins inter-domaines de bout en bout possibles U_k (lignes 2 to 3).

`Étape 2:` pour chaque chemin possible u_k, calculer le délai total δ_k, le coût total σ_k et le niveau minimal de bande passante disponible β_k de chaque lien l_u qui composent le chemin inter-domaine de bout en bout (lignes 8 à 12).

Algorithme 4.1 Global view routing Algorithm (GlobalRT)

```
   Input  : Network G(V, A) and set of flows F
   Output : Ω number of scheduled flows
 1 Ω :=0;
 2 foreach flow k of F do
 3     Compute U_k all possible paths between s_k and d_k;
 4     valid :=0;
 5     i :=0;
 6     while valid=0 do
 7         foreach path u_k of U_k do
 8             foreach link l_u of u_k do
 9                 compute δ_u : total delay of the path;
10                 compute σ_u : total cost of the path;
11                 compute β_u : minimum available bandwidth of end-to-end path l_u of the
                     path U_k;
12             end foreach
13             if β_k≤β_u AND δ_u≤δ_k AND σ_u≤σ_k then
14                 Ω++;
15                 valid :=1;
16             else
17                 valid :=0;
18             end if
19         end foreach
20         valid :=0;
21     end while
22 end foreach
```

Étape 3: vérifier les objectifs de performance requis pour chaque flux k (ligne 13). (β_k,δ_k,σ_k).

Étape 4: si les objectifs de performance requis sont respectés, le flux est considéré comme réussi (ligne 14).

Pour comparer nos résultats à des solutions existantes, nous considérons un algorithme de routage standard (StdRT), qui fonctionne de manière similaire au BRPC Vasseur *et al.* (2009) mais qui a été modifié pour prendre en compte notre objectif de maximiser le nombre de flux acceptés. L'algorithme calcule essentiellement les chemins les plus courts avec des contraintes

Algorithme 4.2 Standard routing Algorithm (StdRT)

```
   Input  : Network G(V, A) and set of flows F
   Output : Ω number of scheduled flows
 1 Ω :=0;
 2 foreach flow k of F do
 3     Compute U_k all possible path between s_k and d_k;
 4     valid :=0;
 5     i :=0;
 6     while valid=0 do
 7         foreach path u_k of U_k do
 8             foreach link l_u of the source domain do
 9                 compute β'_u : minimum available bandwidth of the intra-links source
                    domain;
10             end foreach
11             foreach link l_u of u_k do
12                 compute β_u : minimum available bandwidth of end-to-end path;
13             end foreach
14             if β'_u ≤ β_k then
15                 if β_u ≤ β_k then
16                     Ω++;
17                     valid :=1;
18                 else
19                     valid :=0;
20                 end if
21             end if
22         end foreach
23         valid :=0;
24     end while
25 end foreach
```

dans un réseau multi-domaine sur la base d'une arborescence de chemins entre la source et la destination. L'algorithme 4.2 qui décrit StdRT peut être résumé comme suit :

Étape 1: après la réception du flux k, tous les chemins inter-domaines de bout en bout possibles U_k sont calculés.

Étape 2: pour chaque lien l_u du chemin u_k dans le domaine source, calculer β'_u la bande passante minimale disponible du domaine source intra-liens (lignes 8 à 10).

Étape 3: pour chaque chemin possible u_k, calculer la bande passante minimale disponible β_k pour chaque lien l_u qui compose le chemin de bout en bout(ligne 11 to 13).

Étape 4: vérifier si la bande passante requise β_k du flux k est inférieur à la bande passante disponible des intra-liens du domaine source β'_u et inférieure à la bande passante disponible du chemin de bout en bout β_u (ligne 13). Si oui, le flux est considéré comme réussi (lignes 14 to 21).

4.4 Expérimentations et résultats

Dans cette section, nous présentons les résultats que nous avons trouvé grâce à des simulations pour comparer les solutions proposées, à savoir le modèle ILP, l'algorithme GlobalRT et l'algorithme StdRT.

Nous avons d'abord développé un simulateur en Java pour évaluer les algorithmes GlobalRT et StdRT et nous avons implémenté l'ILP proposé en utilisant le solveur de programmation commerciale CPLEX 12.9.0 offert par IBM (IBM, 2020).

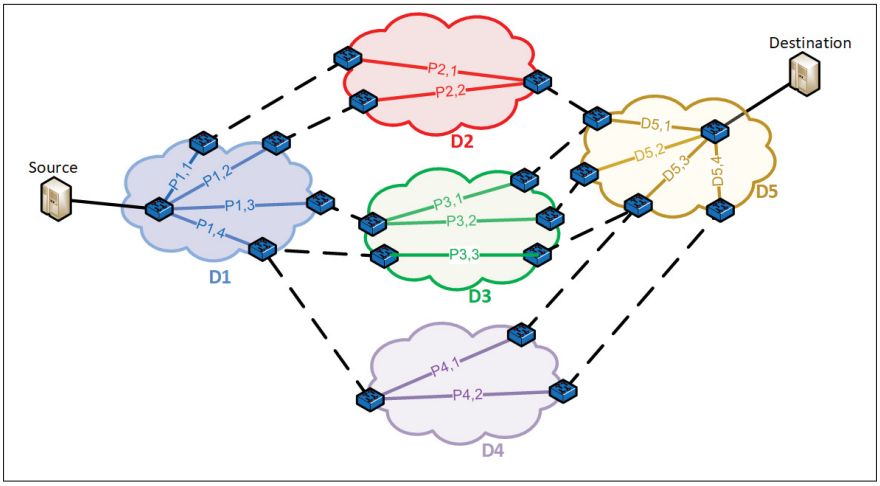


Figure 4.5 Topologie de simulation

Nous avons considéré cinq domaines connectés comme le montre la Figure 4.5. Les domaines sont composés d'un ensemble de chemins intra-domaines $P_{i,j}$, où j est l'identifiant du chemin dans le domaine i. Chaque chemin est caractérisé par la bande passante disponible, le délai total,

le taux de perte et le coût. Dans ce travail, le taux de perte constitue une métrique importante à partager par chaque domaine. Cependant, afin de simplifier les simulations, cette métrique n'est pas considérée car nous supposons que tous les chemins ont le même taux de perte.

Par ailleurs, l'évaluation des performances est effectuée à l'aide de cinq collections différentes de flux de trafic. Chaque collection de flux est générée aléatoirement selon les caractéristiques décrites dans le Tableau 4.3 et nous avons également pris en compte les propriétés de notre topologie comme décrit ci-dessus. Tous les flux ont la même source (domaine D1) et la même destination (domaine D5) comme le montre la Figure 4.5. Nous notons que nous avons considéré un nombre limité de flux afin de réduire le temps de traitement et en supposant que nous pouvons agréger les flux ayant des sources, des destinations et des besoins communs en termes de délai.

Pour chacune des cinq collections de flux, nous évaluons les performances des trois solutions, à savoir le modèle ILP, l'algorithme GlobalRT et l'algorithme StdRT en termes de nombre de flux acceptés, les délais moyens de bout en bout et les coûts monétaires totaux de la transmission des flux.

Tableau 4.3 Caractéristiques des cinq collections de flux considérées

Collection	Nb. flows	BW min	Délai min/max	Coût min/max
C1	25	280	140-180	40-99
C2	58	120	130-180	40-99
C3	100	120	130-180	40-99
C4	1000	140	140-180	40-99
C5	10000	100	130-180	40-99

Nous comparons d'abord le pourcentage des flux réussis pour les trois solutions pour toutes les collections de flux étudiées. La Figure 4.6 montre que l'algorithme GlobalRT accepte jusqu'à 25% plus de flux que l'algorithme de routage standard StdRT. Il fournit également des résultats proches des solutions optimales trouvées avec le modèle ILP. En effet, l'algorithme GlobalRT exploite la vue globale du réseau multi-domaine pour prendre les meilleures décisions de routage par rapport à l'algorithme de routage standard qui a une vue limitée de l'état des domaines.

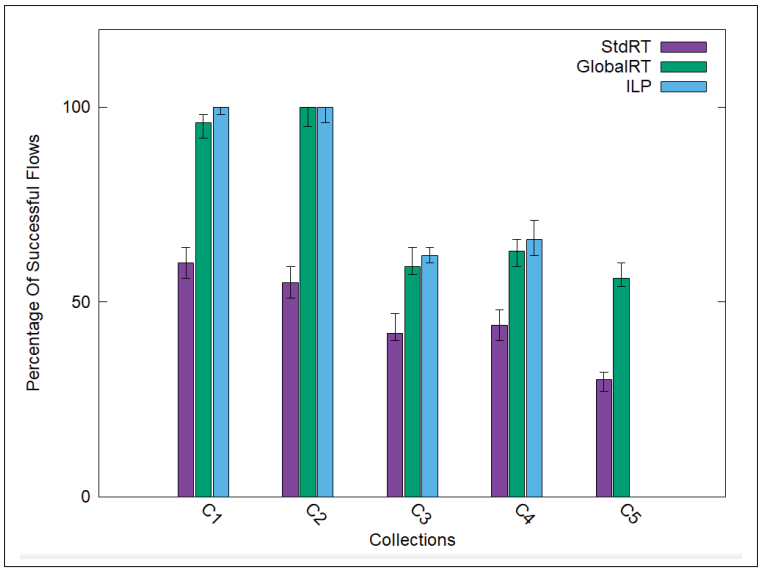


Figure 4.6 Comparaison du pourcentage de flux réussis

De même, la Figure 4.7 démontre que l'algorithme GlobalRT optimise le délai de bout en bout par rapport à l'algorithme de routage standard StdRT. Ces résultats sont légèrement inférieurs à ceux obtenus par le modèle ILP. Les résultats obtenus par le modèle ILP sont similaires à GlobalRT dans le cas des petites et moyennes collections (par exemple, C1, C2, C3 et C4) car l'ILP ne peut pas trouver un résultat optimal dans le cas d'une grande collection (par exemple, C5).

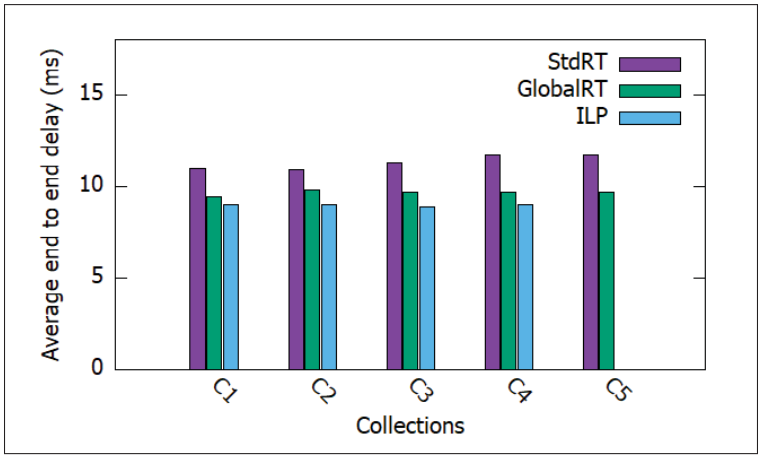


Figure 4.7 Comparaison du délai de bout en bout

Comme le montre la Figure 4.8, nous comparons la fonction de distribution cumulative (CDF) du délai de bout en bout obtenu par chacun des algorithmes étudiés et pour toutes les collections. Nous notons que la valeur la plus basse du délai est de 50 ms, car le chemin le plus court traversant le réseau multi-domaine a un délai de propagation de 50ms et donc il n'y a pas d'autre chemin traversant le réseau qui a un délai inférieur. Cela explique le point d'inflexion vers 50ms. On constate que la performance de l'algorithme GlobalRT avec toutes les collections permet de réduire le délai de bout en bout par rapport à l'algorithme StdRT.

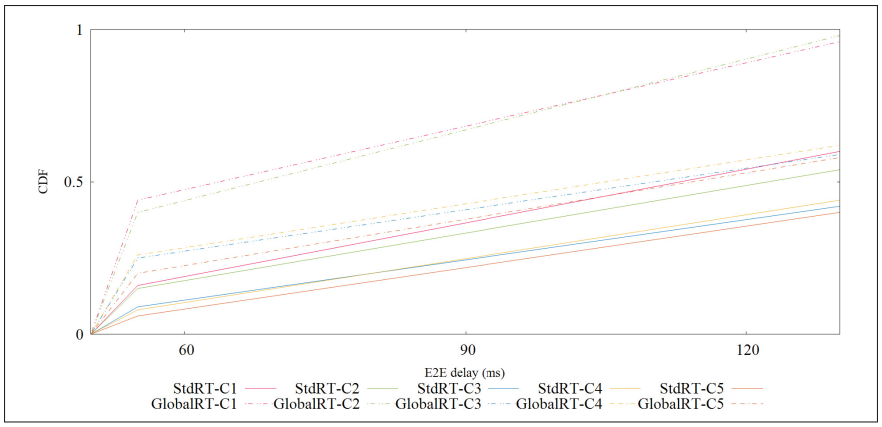


Figure 4.8 Fonction de distribution cumulative du délai de bout en bout

Afin de comparer le coût de transmission des trois solutions (le modèle ILP, l'algorithme GlobalRT et l'algorithme StdRT), nous calculons le coût moyen normalisé par flux. Ce coût normalisé est calculé comme le coût moyen de chaque collection divisé par le résultat obtenu par l'algorithme StdRT (comme référence). La comparaison des résultats du coût moyen normalisé par flux est fournie par la Figure 4.9 et montre que l'algorithme GlobalRT réduit considérablement les coûts par rapport à l'algorithme StdRT, alors qu'il fournit des résultats similaires à ceux obtenus par le modèle ILP.

La Figure 4.10 montre la distribution des revenus (c'est-à-dire revenus gagnés par le fournisseur de services propriétaire du domaine) entre tous les domaines en fonction de l'algorithme et la collection. Les résultats obtenus par l'algorithme StdRT génèrent plus de revenus pour les domaines 1 et 2 et ignorent complètement le domaine 4 (sans revenus). Cependant, la répartition

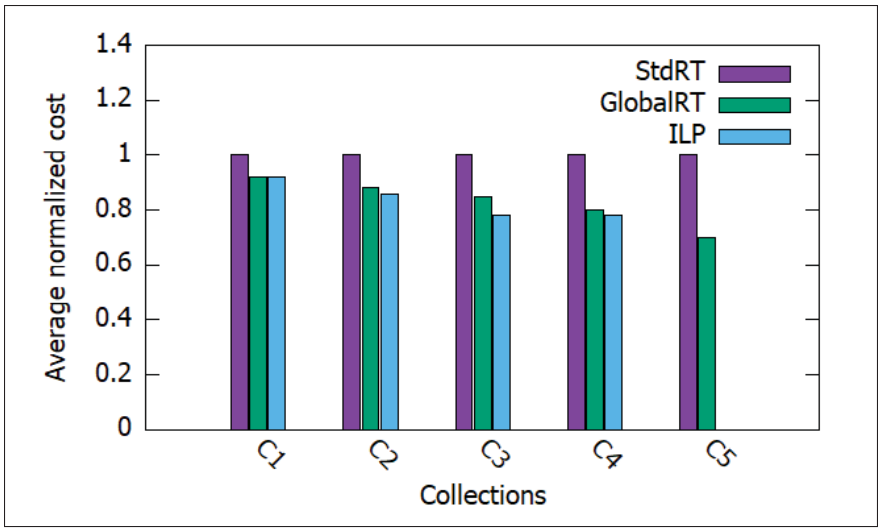


Figure 4.9 Comparaison du coût normalisé moyen par flux

des revenus entre les domaines est modifiée avec l'algorithme GlobalRT en générant des revenus pour le domaine 4, en augmentant les revenus pour le domaine 5 et en diminuant la part des revenus générés par les domaines 1 et 2. Il est à noter que l'algorithme GlobalRT montre des résultats similaires au modèle ILP, sauf pour la collection C5 puisque le modèle ILP ne peut pas trouver un résultat optimal pour des collections ayant un grand nombre de flux.

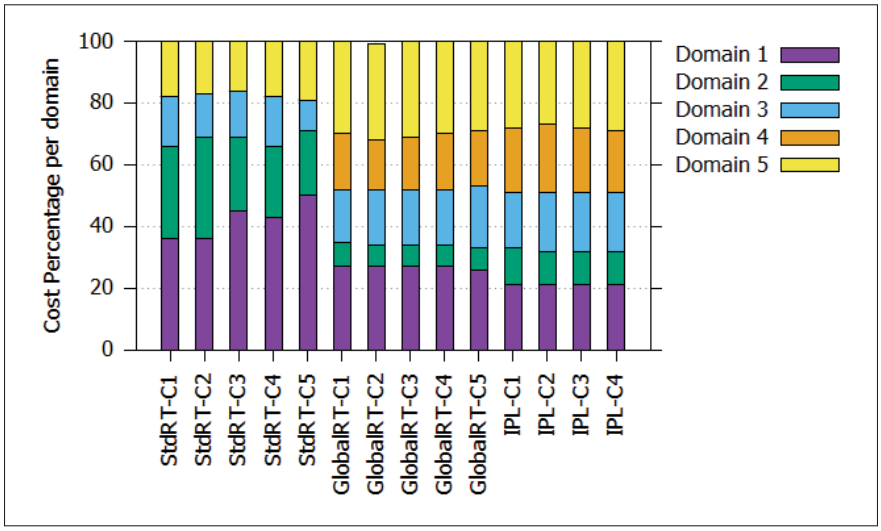


Figure 4.10 Répartition des revenus pour chaque simulation

4.5 Conclusion

Étant donné que les applications du futur exigent des performances rigoureux et des coûts minimaux, il est important de proposer des solutions de routage et de planification de flux qui pourraient offrir un délai et une bande passante garantis de bout en bout avec un coût minimal. Malheureusement, il n'est pas possible de fournir de telles garanties dans Internet d'aujourd'hui, car il est composé de plusieurs domaines administratifs qui sont sous le contrôle de différents opérateurs. Ce travail propose un nouveau mécanisme de routage multi-domaines collaboratif qui tire profit de la collaboration entre les opérateurs de plusieurs domaines de réseau afin de 1) acheminer efficacement les flux entre plusieurs domaines tout en satisfaisant leurs performances de bout en bout et leurs exigences de coût et 2) maximiser le nombre des flux transmis avec succès.

Nous avons d'abord proposé un mécanisme de routage basé sur un courtier pour assurer le routage et la planification des flux sur plusieurs domaines avec des garanties strictes en termes de performances et de coût des flux. Nous avons ensuite formulé le problème comme un programme linéaire entier (IPL) et proposé un algorithme heuristique pour le résoudre pour des scénarios à grande échelle.

Grâce à des simulations approfondies utilisant différents scénarios et configurations, nous avons montré que le mécanisme de routage multi-domaine collaboratif proposé offre une solution quasi-optimale et améliore jusqu'à 25% le nombre de flux transmis à travers plusieurs domaines en répondant à leurs exigences de performance et avec un coût minimal.

CONCLUSION ET RECOMMANDATIONS

La virtualisation des fonctions réseau et la réseautique définie par logiciel sont deux nouveaux paradigmes qui ont été récemment introduites et qui permettent de changer la façon avec laquelle les réseaux sont configurés et maintenus. NFV et SDN offrent plusieurs avantages, notamment la création et la reconfiguration dynamique des VNFs ainsi que le routage dynamique du trafic. Cependant, le déploiement et l'exploitation de ces technologies ne sont pas encore répandus aujourd'hui et beaucoup de défis sont encore à relever pour pouvoir tirer profit des avantages cités. Parmi, ces défis figurent l'allocation des ressources pour les chaînes de services (SFCs) ainsi que la gestion du trafic dans les réseaux SDN multi-domaines.

Dans ce contexte, dans le premier volet de cette thèse, nous avons d'abord effectué une étude détaillée des coûts des machines virtuelles (instances) offertes par Amazon EC2 (Amazon) par rapport à l'emplacement, la taille de l'instance et les performances. Nous avons aussi étudié la possibilité d'offrir les chaînes de fonctions de service en tant que service cloud (SFCaaS). Ainsi, nous nous sommes intéressés au problème d'allocation de ressources aux SFCs dans l'infrastructure physique et le routage de leur trafic. Nous avons formulé le problème en tant qu'un programme linéaire en nombres entiers et proposé deux algorithmes heuristiques visant à maximiser le revenu total du fournisseur de SFCs en tenant compte du coût des instances, le coût d'exploitation opérationnel et le coût de synchronisation entre les instances du VNF.

Ce travail ouvre la porte à d'autres d'opportunités de recherche. Par exemple, il serait intéressant de comparer davantage la performance des VNFs en fonction de la nature de la fonction de réseau implémentée et du hardware qui l'héberge. Une autre piste de recherche consiste à évaluer les coûts de synchronisation entre les instances du même VNF en fonction du type de la fonction à déployer et de l'emplacement des instances.

Dans le deuxième volet de cette thèse, nous nous sommes intéressés à la gestion du trafic dans les réseaux multi-domaines où chaque domaine est administré et géré par un opérateur

de réseau différent. Nous avons donc proposé un nouveau mécanisme de routage collaboratif multi-domaine à base de courtier qui est capable de router efficacement les flux entrants à travers plusieurs domaines tout en garantissant leurs exigences de performance en termes de délai et de bande passante et en maximisant l'utilisation globale du réseau. Nous avons aussi proposé un programme linéaire entier pour résoudre ce problème et avons développé deux algorithmes heuristiques adaptés aux grandes échelles. Les résultats des simulations montrent que les solutions proposées sont capables d'optimiser considérablement l'utilisation du réseau et de maximiser le nombre de flux routés avec des performances garanties.

De nombreux défis restent à relever à l'avenir, notamment l'évolutivité de la solution pour un grand nombre de domaines, la recherche du placement optimal du courtier et l'exploration de la possibilité d'une implémentation distribuée du courtier afin d'accélérer la collecte des données provenant des différents domaines.

BIBLIOGRAPHIE

Addis, B., Belabed, D., Bouet, M. & Secci, S. (2015). Virtual network functions placement and routing optimization. *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, pp. 171–177.

Alomari, Z., Zhani, M. F., Aloqaily, M. & Bouachir, O. (2020, 11). On Minimizing Synchronization Cost in NFV-based Environments. *IEEE/ACM/IFIP International Conference on Network and Service Management (CNSM)*.

Amazon. Amazon EC2 instances. Repéré à https://aws.amazon.com/fr/ec2/instance-types/.

Amazon. (2021). AWS Infrastructure [Web Page]. Repéré à https://infrastructure.aws.

Amazon. (last accessed May 2021). Amazon EC2 instances [Web Page]. Repéré à https://alestic.com/2011/08/ec2-max-instances/.

Amokrane, A., Zhani, M. F., Langar, R., Boutaba, R. & Pujolle, G. (2013). Greenhead : Virtual data center embedding across distributed infrastructures. *IEEE transactions on cloud computing*, 1(1), 36–49.

Ash, J. & Farrel, A. (2006). A path computation element (PCE)-based architecture.

Avallone, S., D'Antonio, S., Esposito, M., Romano, S. P. & Ventre, G. (2006). Resource allocation in multi-domain networks based on service level specifications. *Journal of Communications and Networks*, 8(1), 106-115. doi : 10.1109/JCN.2006.6182910.

Avi. (2020, Novembre, 28). Une introduction aux modèles de services cloud - PaaS, SaaS, IaaS, FaaS et plus encore. . . [Web]. Repéré à https://geekflare.com/fr/cloud-service-models/.

Bari, F., Chowdhury, S. R., Ahmed, R., Boutaba, R. & Duarte, O. C. M. B. (2016). Orchestrating virtualized network functions. *IEEE Transactions on Network and Service Management*, 13(4), 725–739.

Bari, M. F., Chowdhury, S. R., Ahmed, R. & Boutaba, R. (2015a). nf.io : A file system abstraction for NFV orchestration. *IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*.

Bari, M. F., Chowdhury, S. R., Ahmed, R. & Boutaba, R. (2015b). On orchestrating virtual network functions. *2015 11th International Conference on Network and Service Management (CNSM)*, pp. 50–56.

BCS. (2019). History of the cloud [Web Page]. Repéré à https://www.bcs.org/content-hub/history-of-the-cloud/.

Beck, M. T. & Botero, J. F. (2015, Dec). Coordinated Allocation of Service Function Chains. *IEEE Global Communications Conference (GLOBECOM)*, pp. 1-6.

Bednarz, A. (2016). Top reasons for network downtime - Network outages linked to human error, incompatible changes, greater complexity. Repéré le 2016-11-18 à https://www.networkworld.com/article/3142838/top-reasons-for-network-downtime.html.

Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W. & Parulkar, G. (2014). ONOS : towards an open, distributed SDN OS. *Proceedings of the third workshop on Hot topics in software defined networking*, pp. 1–6.

Bhamare, D., Jain, R., Samaka, M. & Erbad, A. (2016). A survey on service function chaining. *Journal of Network and Computer Applications*, 75, 138–155.

Bhamare, D., Samaka, M., Erbad, A., Jain, R., Gupta, L. & Chan, H. A. (2017). Optimal virtual network function placement in multi-cloud service function chaining architecture. *Computer Communications*, 102, 1–16.

Bhardwaj, S., Jain, L. & Jain, S. (2010). Cloud computing : A study of infrastructure as a service (IAAS). *International Journal of engineering and information Technology*, 2(1), 60–63.

Briscoe, G. & Marinos, A. (2009). Digital ecosystems in the clouds : towards community cloud computing. *2009 3rd IEEE international conference on digital ecosystems and technologies*, pp. 103–108.

Cai, D., Wielosz, A. & Wei, S. (2014). Evolve carrier Ethernet architecture with SDN and segment routing. *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, pp. 1–6.

Caizergues, M. (2008). Software defined networking (sdn) definition [Web Page]. Repéré à https://www.reseaux-telecoms.net/actualites/lire-redhat-livre-le-premier-hyperviseur-de-virtualisation-open-source-18385.html.

Carpio, F., Dhahri, S. & Jukan, A. (2017). VNF placement with replication for Loac balancing in NFV networks. *IEEE International Conference on Communications (ICC)*, pp. 1–6.

Casellas, R., Martínez, R., Muñoz, R., Vilalta, R. & Liu, L. (2015a). Control and orchestration of multidomain optical networks with GMPLS as inter-SDN controller communication. *Journal of Optical Communications and Networking*, 7(11), B46–B54.

Casellas, R., Muñoz, R., Martínez, R., Vilalta, R., Liu, L., Tsuritani, T., Morita, I., López, V., de Dios, O. G. & Fernández-Palacios, J. P. (2015b). SDN orchestration of OpenFlow and GMPLS flexi-grid networks with a stateful hierarchical PCE. *IEEE/OSA Journal of Optical*

Communications and Networking, 7(1), A106–A117.

Chang, J., Li, S. & Ranganathan, P. (2020). Compressing and compacting memory on a memory device wherein compressed memory pages are organized by size. Google Patents. US Patent 10,817,178.

Chellappa, R. (1997). Intermediaries in cloud-computing : A new computing paradigm. *INFORMS Annual Meeting, Dallas*, pp. 26–29.

Chen, C., Li, B., Lin, D. & Li, B. (2016). Software-defined inter-domain routing revisited. *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6.

Choi, J. S. & Li, X. (2016). Hierarchical distributed topology discovery protocol for multi-domain SDN networks. *IEEE Communications Letters*, 21(4), 773–776.

Clarke, J., Salgueiro, G. & Pignataro, C. (2016). *Interface to the Routing System (I2RS), Traceability : Framework and Information Model*.

Clayman, S., Maini, E., Galis, A., Manzalini, A. & Mazzocca, N. (2014). The dynamic placement of virtual network functions. *2014 IEEE network operations and management symposium (NOMS)*, pp. 1–9.

Clemm, A., Zhani, M. F. & Boutaba, R. (2020). Network Management 2030 : Operations and Control of Network 2030 Services. *Journal of Network and Systems Management, Springer*.

Cohen, R., Lewin-Eytan, L., Naor, J. S. & Raz, D. (2015). Near optimal placement of virtual network functions. *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 1346–1354.

Douville, R., Rougier, J.-l., Secci, S. et al. (2008). A service plane over the PCE architecture for automatic multidomain connection-oriented services. *IEEE Communications Magazine*, 46(6), 94–102.

Eastep, T. M. (2021). Shorewall Firewall. Accessed : 06-05-2021.

Ekanayake, J. & Fox, G. (2010). High Performance Parallel Computing with Clouds and Cloud Technologies. *Cloud Computing*, pp. 20–38.

El Khoury, N., Ayoubi, S. & Assi, C. (2016). Energy-aware placement and scheduling of network traffic flows with deadlines on virtual network functions. *2016 5th IEEE International Conference on Cloud Networking (Cloudnet)*, pp. 89–94.

Elguea, L. M. & Martinez-Rios, F. (2017). An efficient method to compare latencies in order to obtain the best route for SDN. *Procedia computer science*, 116, 393–400.

Elguea, L. M. & Martinez-Rios, F. (2019). New metrics to modify BGP routes based on SDN. *Wireless Networks*, 1–8.

Enns, R., Bjorklund, M., Schoenwaelder, J. & Bierman, A. (2011). Network configuration protocol (NETCONF).

Famaey, J., Latré, S., Wauters, T. & De Turck, F. (2014). End-to-end resource management for federated delivery of multimedia services. *Journal of network and systems management*, 22(3), 396–433.

Fangzhe, C., Jennifer, R. & Ramesh, V. (2010). Optimal Resource Allocation in Clouds. *Proceedings of the 3rd International Conference on Cloud Computing*, pp. 418–425.

Farrel, A., Vasseur, J. & Ash, G. (2006a). A Path Computation Element (PCE). *RFC 4655*.

Farrel, A., Vasseur, J. & Ash, J. (2006b). RFC 4655 : A Path Computation Element (PCE)-Based Architecture. *IETF, August*.

Feamster, N., Rexford, J., Shenker, S., Clark, R., Hutchins, R., Levin, D. & Bailey, J. (2013). Sdx : A software defined internet exchange. *Open Networking Summit*, 1.

Femmam, M., Kazar, O., Baarir, S. & Fareh, M. E.-k. (2015). A Comparative Study of Levels Scheduling Algorithms in the Cloud Computing. *International Journal of Computing, Communication and Instrumentation Engineering*, 2(2), 196–200.

Filsfils, C., Nainar, N. K., Pignataro, C., Cardona, J. C. & Francois, P. (2015). The Segment Routing Architecture. *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. doi : 10.1109/GLOCOM.2015.7417124.

Filsfils, C., Previdi, S., Decraene, B. & Shakir, R. (2018a). *Resiliency Use Cases in Source Packet Routing in Networking (SPRING) Networks*.

Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S. & Shakir, R. (2018b). *Segment routing architecture*.

Floodlight. (2018). Project Floodlight [Web Page]. Repéré à http://www.projectfloodlight.org/.

Gao, K., Wang, Q. & Xi, L. (2014). Reduct algorithm based execution times prediction in knowledge discovery cloud computing environment. *Int. Arab J. Inf. Technol.*, 11(3), 268–275.

Gartner. (2020a). Software defined networking (sdn) definition [Web Page]. Repéré à https://www.gartner.com/en/newsroom/press-releases/2020-07-23-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-6point3-percent-in-2020.

Gartner. (2020b). Gartner Report : 2020 Magic Quadrant for Cloud Infrastructure and Platform Services [Web Page]. Repéré à https://www.gartner.com/en/documents/3989743/magic-quadrant-for-cloud-infrastructure-and-platform-ser.

Gartner. (2020c). Gartner Forecasts Worldwide Public Cloud Revenue to Grow 6.3% in 2020 [Web Page]. Repéré à https://www.gartner.com/en/newsroom/press-releases/2020-07-23-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-6point3-percent-in-2020.

Gartner. (2021). Community Cloud [Web Page]. Repéré à https://www.gartner.com/en/information-technology/glossary/community-cloud.

Gember, A., Akella, A., Anand, A., Benson, T. & Grandl, R. (2012). Stratos : Virtual middleboxes as first-class entities.

Gerola, M., Lucrezia, F., Santuari, M., Salvadori, E., Ventre, P. L., Salsano, S. & Campanella, M. (2016). ICONA : A Peer-to-Peer Approach for Software Defined Wide Area Networks Using ONOS. *2016 Fifth European Workshop on Software-Defined Networks (EWSDN)*, pp. 37–42.

Ghanbari, H., Simmons, B., Litoiu, M. & Iszlai, G. (2012). Feedback-based optimization of a private cloud. *Future Generation Computer Systems*, 28(1), 104–111.

Ghaznavi, M., Khan, A., Shahriar, N., Alsubhi, K., Ahmed, R. & Boutaba, R. (2015). Elastic virtual network function placement. *IEEE 4th International Conference on Cloud Networking (CloudNet)*.

Ghaznavi, M., Shahriar, N., Ahmed, R. & Boutaba, R. (2016). Service function chaining simplified. *arXiv preprint arXiv :1601.00751*.

Ghrada, N. (2018). Cost-aware VNF placement in cloud infrastructures. *École de technologie supérieure*.

Ghrada, N., Zhani, M. F. & Elkhatib, Y. (2018, 25-29,). Price and Performance of Cloud-hosted Virtual Network Functions : Analysis and Future Challenges. *IEEE Performance Issues in Virtualized Environments and Software Defined Networking (PVE-SDN NetSoft 2018)*.

Giorgetti, A., Paolucci, F., Cugini, F. & Castoldi, P. (2011, March). Hierarchical PCE in GMPLS-based multi-domain Wavelength Switched Optical Networks. *2011 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference*, pp. 1-3.

Giorgetti, A. (2015). Proactive H-PCE architecture with BGP-LS update for multidomain elastic optical networks. *Journal of Optical Communications and Networking*, 7(11), B1–B9.

González de Dios, O., Casellas, R., Morro, R., Paolucci, F., López, V., Martínez, R., Muñoz, R., Vilalta, R. & Castoldi, P. (2015, March). First multi-partner demonstration of BGP-LS enabled inter-domain EON control with H-PCE. *2015 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1-3. doi : 10.1364/OFC.2015.Th1A.4.

Google. (2021). Google Cloud [Web Page]. Repéré à https://cloud.google.com/.

Goyal, S. (2014). Public vs private vs hybrid vs community-cloud computing : a critical review. *International Journal of Computer Network and Information Security*, 6(3), 20.

Gros, M. (2020, Aout, 12). Le marché de l'IaaS a crû de 37,3% à 44,5 Md$ en 2019 [Article]. Repéré à https://www.lemondeinformatique.fr/actualites/lire-le-marche-de-l-iaas-a-cru-de-37-3-a-44-5-md\protect\T1\textdollar-en-2019-80017.html.

Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N. & Shenker, S. (2008). NOX : towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3), 105–110.

Gupta, A., Goswami, P., Chaudhary, N. & Bansal, R. (2020). Deploying an Application using Google Cloud Platform. *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pp. 236–239.

Gupta, L., Samaka, M., Jain, R., Erbad, A., Bhamare, D. & Metz, C. (2017). COLAP : A predictive framework for service function chain placement in a multi-cloud environment. *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1–9.

Halpern, J., Salim, J. H. et al. (2010). *Forwarding and control element separation (ForCES) forwarding element model*.

Halpern, J., Pignataro, C. et al. (2015). Service function chaining (sfc) architecture. Dans *RFC 7665*.

Hamrén, O. (2012). Mobile phones and cloud computing : A quantitative research paper on mobile phone application offloading by cloud computing utilization.

Herrera, J. G. & Botero, J. F. (2016). Resource allocation in NFV : A comprehensive survey. *IEEE Transactions on Network and Service Management*, 13(3), 518–532.

Huang, Q.-y. & Huang, T.-l. (2010). An optimistic job scheduling strategy based on QoS for cloud computing. *2010 International Conference on Intelligent Computing and Integrated Systems*, pp. 673–675.

IBM. (2020). User's Manual for CPLEX : V12.9.0 [Web Page]. Repéré à https://www.ibm.com/support/knowledgecenter/en/SSSA5P_12.9.0/ilog.odms.cplex.help/CPLEX/homepages/usrmancplex.html.

IBM Cloud Education. (2019, Juillet, 12). IaaS (Infrastructure-as-a-Service) [Article]. Repéré à https://www.ibm.com/cloud/learn/iaas.

Ifrah, S. (2021). Get Started with Google Cloud Platform (GCP). Dans *Getting Started with Containers in Google Cloud Platform* (pp. 1–37). Springer.

Initiative, E. C. R. et al. (2008). Energy efficiency for Network Equipment : Two steps beyond greenwashing. *White Paper, August*, 10.

It-connect. (2021). Les types d'hyperviseurs. Repéré le 2012-11-15 à https://www.it-connect.fr/les-types-dhyperviseurs/.

Jahan, R., Shaik, S., Kotaru, K., Sangam, S. & Kuppili, D.-C. (2018). OpenDaylight Project [Web Page]. Repéré à https://wiki.opendaylight.org/.

Jansen, W. A., Grance, T. et al. (2011). Guidelines on security and privacy in public cloud computing.

Janz, C., Ong, L., Sethuraman, K. & Shukla, V. (2016). Emerging transport SDN architecture and use cases. *IEEE Communications Magazine*, 54(10), 116-121. doi : 10.1109/MCOM.2016.7588279.

Jeba, J. A., Roy, S., Rashid, M. O., Atik, S. T. & Whaiduzzaman, M. (2019). Towards green cloud computing an algorithmic approach for energy minimization in cloud data centers. *International Journal of Cloud Applications and Computing (IJCAC)*, 9(1), 59–81.

John, W., Moradi, F., Pechenot, B. & Sköldström, P. (2017). Meeting the observability challenges for VNFs in 5G systems. *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 1127–1130.

Joshi, K. D. & Kataoka, K. (2019). PRIME-Q : Privacy Aware End-to-End QoS Framework in Multi-Domain SDN. *2019 IEEE Conference on Network Softwarization (NetSoft)*, pp. 169–177.

Jrad, F., Tao, J. & Streit, A. (2013). A broker-based framework for multi-cloud workflows. *Proceedings of the 2013 international workshop on Multi-cloud applications and federated*

clouds, pp. 61–68.

Kamongi, P. (2019). Cloud Security and Privacy Management. *Security, Privacy, and Digital Forensics in the Cloud*, 109.

Karakus, M. & Durresi, A. (2015, March). A Scalable Inter-AS QoS Routing Architecture in Software Defined Network (SDN). *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, pp. 148-154. doi : 10.1109/AINA.2015.179.

Karakus, M. & Durresi, A. (2017). Quality of service (QoS) in software defined networking (SDN) : A survey. *Journal of Network and Computer Applications*, 80, 200–218.

Katsalis, K., Rofoee, B., Landi, G., Riera, J. F., Kousias, K., Anastasopoulos, M., Kiraly, L., Tzanakaki, A. & Korakis, T. (2017). Implementation experience in multi-domain SDN : Challenges, consolidation and future directions. *Computer Networks*, 129, 142–158.

Kelly, M., Furey, E. & Curran, K. (2021). How to Achieve Compliance with GDPR Article 17 in a Hybrid Cloud Environment. *Sci*, 3(1), 3.

Kempf, J., Körling, M., Baucke, S., Touati, S., McClelland, V., Más, I. & Bäckman, O. (2014, June). Fostering rapid, cross-domain service innovation in operator networks through Service Provider SDN. *2014 IEEE International Conference on Communications (ICC)*, pp. 3064-3069. doi : 10.1109/ICC.2014.6883791.

Kessaci, Y. (2013). *Multi-criteria scheduling on clouds*. (Thèse de doctorat, Université des Sciences et Technologie de Lille-Lille I).

Kim, S., Han, Y. & Park, S. (2016). An energy-aware service function chaining and reconfiguration algorithm in NFV. *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W)*, pp. 54–59.

King, D. & Farrel, A. (2012). The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS. *IETF, RFC6805*.

Kotronis, V., Dimitropoulos, X. & Ager, B. (2012). Outsourcing the routing control logic : better internet routing based on SDN principles. *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, pp. 55–60.

Kotronis, V., Gämperli, A. & Dimitropoulos, X. (2015). Routing centralization across domains via SDN : A model and emulation framework for BGP evolution. *Computer Networks*, 92, 227–239.

Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S. & Uhlig, S. (2015). Software-defined networking : A comprehensive survey. *Proceedings of the IEEE*, 103(1), 14–76.

Krishnaswamy, D., Kothari, R. & Gabale, V. (2015). Latency and policy aware hierarchical partitioning for nfv systems. *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, pp. 205–211.

Kumar, R., Hasan, M., Padhy, S., Evchenko, K., Piramanayagam, L., Mohan, S. & Bobba, R. B. (2017a, Dec). End-to-End Network Delay Guarantees for Real-Time Systems Using SDN. *2017 IEEE Real-Time Systems Symposium (RTSS)*, pp. 231-242. doi : 10.1109/RTSS.2017.00029.

Kumar, R., Hasan, M., Padhy, S., Evchenko, K., Piramanayagam, L., Mohan, S. & Bobba, R. B. (2017b, Dec). End-to-End Network Delay Guarantees for Real-Time Systems Using SDN. *2017 IEEE Real-Time Systems Symposium (RTSS)*, pp. 231-242. doi : 10.1109/RTSS.2017.00029.

Labovitz, C., Ahuja, A., Bose, A. & Jahanian, F. (2000). Delayed Internet routing convergence. *ACM SIGCOMM Computer Communication Review*, 30(4), 175–187.

Labovitz, C., Ahuja, A., Bose, A. & Jahanian, F. (2001). Delayed Internet routing convergence. *IEEE/ACM transactions on networking*, 9(3), 293–306.

Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J. & Jahanian, F. (2011). Internet inter-domain traffic. *ACM SIGCOMM Computer Communication Review*, 41(4), 75–86.

Lacaze, G. (2013). Variation lexicologique dans les traductions de l'expression cloud computing dans le journal Le Monde : vers une recherche de transparence. *ASp. la revue du GERAS*, (63), 55–73.

Li, D., Hong, P., Xue, K. & Pei, J. (2019). Availability aware VNF deployment in datacenter through shared redundancy and multi-tenancy. *IEEE Transactions on Network and Service Management*, 16(4), 1651–1664.

Lin, P., Hart, J., Krishnaswamy, U., Murakami, T., Kobayashi, M., Al-Shabibi, A., Wang, K.-C. & Bi, J. (2013). Seamless interworking of SDN and IP. *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pp. 475–476.

Lin, P., Bi, J., Wolff, S., Wang, Y., Xu, A., Chen, Z., Hu, H. & Lin, Y. (2015). A west-east bridge based SDN inter-domain testbed. *IEEE Communications Magazine*, 53(2), 190–197.

Luizelli, M. C., Bays, L. R., Buriol, L. S., Barcellos, M. P. & Gaspary, L. P. (2015). Piecing together the NFV provisioning puzzle : Efficient placement and chaining of virtual network functions. *2015 IFIP/IEEE International Symposium on Integrated Network Management*

(IM), pp. 98–106.

Lukovszki, T., Rost, M. & Schmid, S. (2016). It's a match ! near-optimal and incremental middlebox deployment. *ACM SIGCOMM Computer Communication Review*, 46(1), 30–36.

Ma, W., Beltran, J., Pan, Z., Pan, D. & Pissinou, N. (2017). SDN-based traffic aware placement of NFV middleboxes. *IEEE Transactions on Network and Service Management*, 14(3), 528–542.

Manolova, A. & Ruepp, S. (2010, March). Export policies for multi-domain WDM networks. *2010 Conference on Optical Fiber Communication (OFC/NFOEC), collocated National Fiber Optic Engineers Conference*, pp. 1-3. doi : 10.1364/NFOEC.2010.NWA5.

Marconett, D. & Yoo, S. B. (2015). Flowbroker : A software-defined network controller architecture for multi-domain brokering and reputation. *Journal of Network and Systems Management*, 23(2), 328–359.

Marinos, A. & Briscoe, G. (2009). Community Cloud Computing. *Cloud Computing*, pp. 472–484.

MarketsandMarkets. (2018). *Hybrid Cloud Market by Component, Service Type (Cloud Management and Orchestration, Disaster Recovery, and Hybrid Hosting), Service Model, Organization Size (SMEs and Large Enterprises), Vertical, and Region - Global Forecast to 2023* (Rapport n°154). https ://www.marketsandmarkets.com/ : MarketsandMarkets.

Martin, L. (2020). Cloud Computing, Smart Technology, and Library Automation. Dans *Emerging Trends and Impacts of the Internet of Things in Libraries* (pp. 105–123). IGI Global.

McKeown, N. (2011). How SDN will shape networking. *Open Networking Summit*.

Mechtri, M., Ghribi, C. & Zeghlache, D. (2016). A scalable algorithm for the placement of service function chains. *IEEE Transactions on Network and Service Management*, 13(3), 533–546.

Medved, J., Varga, R., Tkacik, A. & Gray, K. (2014). Opendaylight : Towards a model-driven sdn controller architecture. *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, pp. 1–6.

Mell, P. & Grance, T. (2009). Effectively and securely using the cloud computing paradigm. *NIST, Information Technology Laboratory*, 2(8), 304–311.

Mell, P., Grance, T. et al. (2011). The NIST definition of cloud computing.

Metz, C. (2001). Interconnecting ISP networks. *IEEE Internet Computing*, 5(2), 74-80. doi : 10.1109/4236.914650.

Microsoft. (2021). Qu'est-ce que le SaaS ? [Web Page]. Repéré à https://azure.microsoft.com/fr-ca/overview/what-is-saas/.

Mijumbi, R., Serrat, J., Gorricho, J.-L., Bouten, N., De Turck, F. & Davy, S. (2015). Design and evaluation of algorithms for mapping and scheduling of virtual network functions. *Proceedings of the 2015 1st IEEE conference on network softwarization (NetSoft)*, pp. 1–9.

Mininet. (2018). Mininet [Web Page]. Repéré à http://mininet.org.

Mirjalily, G. & Zhiquan, L. (2018). Optimal network function virtualization and service function chaining : A survey. *Chinese Journal of Electronics*, 27(4), 704–717.

Moufakir, T., Zhani, M. F., Gherbi, A. & Bouachir, O. (2022). Collaborative Multi-domain Routing in SDN Environments. *Journal of Network and Systems Management*, 30(1), 1–23.

Murukan, P., Jamaluddine, D., Kolhapure, S., Mikhael, F. & Nouzari, S. (2016). A Cost-based Placement Algorithm for Multiple Virtual Security Appliances in Cloud using SDN : MO-UFLP (Multi-Ordered Uncapacitated Facility Location Problem). *arXiv preprint arXiv :1602.08155*.

Mydyti, H., Ajdari, J. & Zenuni, X. (2020). Cloud-based Services Approach as Accelerator in Empowering Digital Transformation. *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, pp. 1390–1396.

Nandan, A., Porowski, M. C., Liu, P. & Forsman, J. (2018). Survey Analysis : NFV/SDN Adoption in CSPs Calls for Strategic Changes in Transformation Programs. Gartner.

Nayyer, A., Sharma, A. K. & Awasthi, L. K. (2019). Laman : A supervisor controller based scalable framework for software defined networks. *Computer Networks*, 159, 125–134.

NIST, N. I. o. S. & Technology. (2011, Septembre). The NIST Definition of Cloud Computing. Repéré le 2021-01-03 à https://csrc.nist.gov/publications/detail/sp/800-145/final.

Ohri, A. (2014). *R for cloud computing : An approach for data scientists*. Springer.

Olorunfemi Abe, J. & Ali Mantar, H. (2017, Nov). Multipath routing and brokering in inter-domain or inter-as with SDN : A model. *2017 Advances in Wireless and Optical Communications (RTUWO)*, pp. 192-197. doi : 10.1109/RTUWO.2017.8228532.

ONF. (2018). Open Networking Foundation (ONF) [Web Page]. Repéré à https://www.opennetworking.org/.

ONF. (2020). Software defined networking (sdn) definition [Web Page]. Repéré à https://www.opennetworking.org/.

OQLF, O. q. d. l. l. f. (2015). Infonuagique. Repéré le 2021-01-03 à http://gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx?Id_Fiche=26501384.

Paolucci, F., Cugini, F., Valcarenghi, L. & Castoldi, P. (2008, Feb). Enhancing Backward Recursive PCE-based Computation (BRPC) for Inter-Domain Protected LSP Provisioning. *OFC/NFOEC 2008 - 2008 Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference*, pp. 1-3. doi : 10.1109/OFC.2008.4528564.

Paolucci, F., Cugini, F., Giorgetti, A., Sambo, N. & Castoldi, P. (2013). A survey on the path computation element (PCE) architecture. *IEEE Communications Surveys & Tutorials*, 15(4), 1819–1841.

Patel, A., Vutukuru, M. & Krishnaswamy, D. (2017). Mobility-aware VNF placement in the LTE EPC. *2017 IEEE conference on network function virtualization and software defined networks (NFV-SDN)*, pp. 1–7.

Pham, C., Tran, N. H., Ren, S., Saad, W. & Hong, C. S. (2017). Traffic-aware and energy-efficient vNF placement for service chaining : Joint sampling and matching approach. *IEEE Transactions on Services Computing*, 13(1), 172–185.

Phan, X. T., Thoai, N. & Kuonen, P. (2013). A collaborative model for routing in multi-domains OpenFlow networks. *International Conference on Computing, Management and Telecommunications (ComManTel)*, pp. 278–283.

Racheg, W., Ghrada, N. & Zhani, M. F. (2017a). Profit-driven resource provisioning in NFV-based environments. *IEEE Conf. on Communications (ICC)*. doi : 10.1109/ICC.2017.7997163.

Racheg, W., Ghrada, N. & Zhani, M. F. (2017b). Profit-driven resource provisioning in NFV-based environments. *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7.

Rankothge, W., Le, F., Russo, A. & Lobo, J. (2015a). Experimental results on the use of genetic algorithms for scaling virtualized network functions. *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, pp. 47–53.

Rankothge, W., Ma, J., Le, F., Russo, A. & Lobo, J. (2015b). Towards making network function virtualization a cloud computing service. *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 89–97.

Rankothge, W., Le, F., Russo, A. & Lobo, J. (2017). Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms. *IEEE Transactions on Network and Service Management*, 14(2), 343–356.

Rashid, A. & Chaturvedi, A. (2019). Cloud computing characteristics and services : a brief review. *International Journal of Computer Sciences and Engineering*, 7(2), 421–426.

Redhat. (2021). Qu'est-ce que le FaaS ? [Web]. Repéré à https://www.redhat.com/fr/topics/cloud-native-apps/what-is-faas.

Reese, G. (2009). *Cloud application architectures : building applications and infrastructure in the cloud*. " O'Reilly Media, Inc.".

Rittinghouse, J. W. & Ransome, J. F. (2016). *Cloud computing : implementation, management, and security*. CRC press.

Santos, G. L., Bezerra, D. d. F., Rocha, É. d. S., Ferreira, L., Moreira, A. L. C., Gonçalves, G. E., Marquezini, M. V., Recse, Á., Mehta, A., Kelner, J. et al. (2022). Service Function Chain Placement in Distributed Scenarios : a Systematic Review. *Journal of Network and Systems Management*, 30(1), 1–39.

Sarna, D. E. (2010). *Implementing and developing cloud computing applications*. CRC Press.

Sezer, S., Scott-Hayward, S., Chouhan, P. K., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M. & Rao, N. (2013). Are we ready for SDN ? Implementation challenges for software-defined networks. *IEEE Communications Magazine*, 51(7), 36-43. doi : 10.1109/MCOM.2013.6553676.

Shenker, S., Casado, M., Koponen, T., McKeown, N. et al. (2011). The future of networking, and the past of protocols. *Open Networking Summit*, 20, 1–30.

Sherwood, R., Gibb, G., Yap, K.-K., Appenzeller, G., Casado, M., McKeown, N. & Parulkar, G. (2009). Flowvisor : A network virtualization layer. *OpenFlow Switch Consortium, Tech. Rep*, 1, 132.

Siracusa, D., Grita, S., Maier, G., Pattavina, A., Paolucci, F., Cugini, F. & Castoldi, P. (2012). Domain sequence protocol (DSP) for PCE-based multi-domain traffic engineering. *IEEE/OSA Journal of Optical Communications and Networking*, 4(11), 876-884. doi : 10.1364/JOCN.4.000876.

Snort. (2021). Snort - Network Intrusion and Prevention System. Accessed : 06-05-2021, Repéré à https://www.snort.org/.

Stallings, W. (2015). *Foundations of modern networking : SDN, NFV, QoE, IoT, and Cloud*. Addison-Wesley Professional.

Statista. (2020). Public cloud services : global market forecast 2009-2014 [Web Page]. Repéré à https://www.statista.com/statistics/203603/forecast-for-the-worldwide-public-cloud-services-market-until-2014/.

Tajiki, M. M., Salsano, S., Chiaraviglio, L., Shojafar, M. & Akbari, B. (2018). Joint energy efficient and QoS-aware path allocation and VNF placement for service function chaining. *IEEE Transactions on Network and Service Management*, 16(1), 374–388.

Tashtarian, F., Zhani, M. F., Fatemipour, B. & Yazdani, D. (2020). CoDeC : A Cost-Effective and Delay-Aware SFC Deployment. *IEEE Transactions on Network and Service Management*, 17(2), 793-806.

Thai, P. & de Oliveira, J. C. (2013). Decoupling policy from routing with software defined interdomain management : Interdomain routing for SDN-based networks. *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–6.

Tibco. (2021a). What is BaaS ? | Backend-as-a-Service vs. serverless [Web]. Repéré à https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/.

Tibco. (2021b). What is Data as a Service (DaaS) ? [Web]. Repéré à https://www.tibco.com/reference-center/what-is-daas.

Tootoonchian, A. & Ganjali, Y. (2010). Hyperflow : A distributed control plane for openflow. *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, 3.

Vaquero, L. M., Rodero-Merino, L., Caceres, J. & Lindner, M. (2008). A break in the clouds : towards a cloud definition. ACM New York, NY, USA.

Vaquero, L. M., Cuadrado, F., Elkhatib, Y., Bernal-Bernabe, J., Srirama, S. N. & Zhani, M. F. (2019). Research challenges in nextgen service orchestration. *Future Generation Computer Systems*, 90, 20–38.

Varghese, B., Leitner, P., Ray, S., Chard, K., Barker, A., Elkhatib, Y., Herry, H., Hong, C., Singer, J., Tso, F. P., Yoneki, E. & Zhani, M. F. (2019). Cloud Futurology. *IEEE Computer*, 52(9), 68-77.

Vasseur, J., Zhang, R., Bitar, N. & Le Roux, J. (2009). *A backward-recursive PCE-based computation (BRPC) procedure to compute shortest constrained inter-domain traffic engineering label switched paths*.

Veronica, H. (2012). Benefits of Private Cloud over Public Cloud [Web Page]. Repéré à https://www.datamation.com/cloud-computing/benefits-of-private-cloud-over-public-cloud-2.html.

Vilalta, R., Mayoral, A., Pubill, D., Casellas, R., Martínez, R., Serra, J., Verikoukis, C. & Muñoz, R. (2016). End-to-end SDN orchestration of IoT services using an SDN/NFV-enabled edge node. *2016 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1–3.

Wada, I. (2018). Cloud computing implementation in libraries : A synergy for library services optimization. *International journal of library and Information Science*, 10(2), 17–27.

Wang, X., Wu, C., Le, F., Liu, A., Li, Z. & Lau, F. (2016). Online vnf scaling in datacenters. *Cloud Computing (CLOUD), 2016 IEEE 9th International Conference on*, pp. 140–147.

Xie, Y., Liu, Z., Wang, S. & Wang, Y. (2016). Service function chaining resource allocation : A survey. *arXiv preprint arXiv :1608.00095*.

Xu, Z., Zhang, X., Yu, S. & Zhang, J. (2018). Energy-efficient virtual network function placement in telecom networks. *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7.

Yadegaridehkordi, E., Nilashi, M., Shuib, L., Asadi, S. & Ibrahim, O. (2019). Development of a SaaS adoption decision-making model using a new hybrid MCDM approach. *International Journal of Information Technology & Decision Making*, 18(06), 1845–1874.

Yadegaridehkordi, E., Nilashi, M., Shuib, L. & Samad, S. (2020). A behavioral intention model for SaaS-based collaboration services in higher education. *Education and information technologies*, 25(2), 791–816.

Yang, K., Zhang, H. & Hong, P. (2016). Energy-aware service function placement for service function chaining in data centers. *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6.

Yin, H., Xie, H., Tsou, T., Lopez, D., Aranda, P. & Sidi, R. (2012). SDNi : A Message Exchange Protocol for Software Defined Networks (SDNs) across Multiple Domains. *IETF draft*.

Zhang, Q., Cheng, L. & Boutaba, R. (2010). Cloud computing : state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1), 7–18.

Zhang, X., Wu, C., Li, Z. & Lau, F. C. (2017). Proactive VNF provisioning with multi-timescale cloud resources : Fusing online learning and online optimization. *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9.

Zhani, M. F. & Elbakoury, H. (2020). FlexNGIA : A Flexible Internet Architecture for the Next-Generation Tactile Internet. *Journal of Network and Systems Management, Springer*.

Éditorial Geekflare. (2020). Une introduction aux modèles de services cloud - PaaS, SaaS, IaaS, FaaS et plus encore. Repéré le 2020-11-28 à https://geekflare.com/fr/cloud-service-models/.