HAL
open science

Collaborative Content Distribution over a VNF-as-a-Service platform

Nicolas Herbaut

To cite this version:

Nicolas Herbaut. Collaborative Content Distribution over a VNF-as-a-Service platform. Networking and Internet Architecture [cs.NI]. Université de Bordeaux, 2017. English. NNT : 2017BORD0738 . tel-01668553

HAL Id: tel-01668553

https://theses.hal.science/tel-01668553

Submitted on 20 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE
SPÉCIALITÉ INFORMATIQUE

Collaborative Content Distribution
Over a VNF-as-a-Service Platform

par

Nicolas HERBAUT

Soutenue le 13/11/2017, devant le jury composé de :

Présidente du jury
Francine KRIEF, Professor ..... Bordeaux-INP, LaBRI UMR 5800, France

Directeur de Thèse
Daniel NEGRU, Associate Professor, HDR ..... Bordeaux-INP, LaBRI UMR 5800, France

Rapporteurs
Laurent MATHY, Professor ..... University of Liège, Belgium
Tarik TALEB, Professor ..... Aalto University, Finland

Examinateurs
Liam MURPHY, Professor ..... University College of Dublin, Ireland
Adlen KSENTINI, Associate Professor, HDR ..... Eurecom, France
Marko VUKOLIĆ, Research Staff Member ..... IBM Research, Switzerland

Collaborative Content Distribution over a VNF-as-a-Service platform

Abstract

The constant rise of Over-The-Top video consumption nowadays challenges the current Internet architecture. As an overwhelming majority of the bandwidth today is dedicated to the delivery of video contents, Internet actors such as Content Providers, Content Delivery Network, Internet Service Providers are forced to optimize their networks to support the Quality of Experience expected by the End-Users . Such costly specialized networks participate to the so-called "ossification" of the Internet which makes the architecture harder to update, as the placement and dimensioning of links and middleboxes may be hard to change in the future to support new use cases. Fortunately, the trend of Virtualizating and Softwarizing the networks pushed by major Telco operators, vendors and standardization bodies has given hopes that the computing and networking infrastructure can be easily re-purposed. The agility promoted by technologies such as Network Function Virtualization and Software Defined Networking makes it possible for middleboxes to be deployed as Virtual Network Functions that can run on "commercial off-the-shelf" hardware while having the network managed by a logically centralized controller deploying network configurations on programmable forwarding devices.

The goal of this thesis is to show how content distribution can be collaboratively improved thanks to Network Softwarization. First, we propose CDN-as-a-Service (CDNaaS), a complete solution to virtualize a Content Delivery Network on top of a VNF-as-a-Service platform, adopted and evaluated at large scale in the FP7 T-Nova European project. We elaborate on the interfaces, architecture and design choices made to implement the platform to support performance, automatic scaling and re-usability. Then, we propose two collaboration models allowing the content delivery actors to work jointly on improving End-User Quality of experience while fostering a healthy competition and a fair balance of revenue. Finally, we study the challenges of NFV resource allocation for the vCDN service and propose several heuristics and algorithms to optimize the proposed solution in a cost-effective way. This thesis paves the way towards a collaborative content distribution allowing End-Users to access their content with the highest standards while contributing to a sound development of the Internet.

Keywords: *NFV, SDN, Content Delivery, Blockchain*

présentée au Laboratoire Bordelais de Recherche en Informatique — UMR 5800
LaBRI — 351, Avenue de la Libération— 33400 Talence — France

Distribution de contenus collaborative basée sur une plateforme de fonctions réseaux virtualisées en tant que service

Résumé

L'augmentation constante de la consommation de vidéos par des services par contournement (Over-The-Top) met à mal l'architecture actuelle d'Internet. Alors qu'une écrasante majorité de la bande passante aujourd'hui est allouée à la livraison de contenus, les acteurs de l'Internet, tels que les fournisseurs de contenus, les réseaux de distribution de contenus et les fournisseurs d'accès sont obligés d'optimiser leurs réseaux pour supporter la qualité d'expérience attendue par l'utilisateur final. Ces réseaux couteux et spécialisés participent à l'ossification de l'Internet, rendant l'évolution de son architecture plus difficile à moyen terme. En effet, un choix trop spécifique de dimensionnement des liens et de la localisation des middle-boxes peut être un frein à une mise à jour ultérieure en vue du support de nouveaux cas d'utilisation. Heureusement, les technologies de virtualisation récemment promues par les grands opérateurs Internet, les vendeurs de solutions et les organismes de standardisation, permettent une réelle programmabilité du réseau et une plus grande versatilité dans les usages de nouveaux équipements. En effet, l'agilité apportée par ces technologies permet le déploiement de Fonctions Réseaux Virtuelles (VNF, Virtual Network Functions) pouvant s'exécuter sur des serveurs de série à bas coût. Quant au Software-Defined Networking , il rend possible une gestion du réseau logiquement centralisée permettant la programmation des commutateurs.

L'objectif de cette thèse est de montrer comment la distribution de contenu peut être améliorée collaborativement à l'aide de la programmabilité de réseaux. Tout d'abord, nous proposons CDNaaS, une solution complète de réseau de livraison de contenu déployée sur une plateforme de "fonctions réseaux en tant que service" adoptée et évaluée à large échelle dans le cadre du projet Européen FP7 T-NOVA. Nous précisons les interfaces, l'architecture et les choix de conception fait pour développer la plateforme afin de fournir performance, auto-dimensionnement et réutilisabilité. Puis, nous proposons deux modèles de collaboration permettant aux acteurs de la livraison de contenu de travailler ensemble afin d'augmenter la qualité d'expérience pour l'utilisateur final, tout en promouvant une compétition saine et une répartition équilibrée de la valeur ajoutée. Finalement, nous étudions les défis liés à l'allocation de ressources virtuelles dans le cas d'un service vCDN, et proposons plusieurs heuristiques et algorithmes permettant l'optimisation du coût du service.

Cette thèse ouvre la voie à une distribution de contenu collaborative permettant aux utilisateurs d'accéder à leurs contenus avec un haut standard de qualité, tout en contribuant à un développement sain de l'Internet.

Mots clés : *NFV, SDN, Livraison de contenu, Blockchain*

Remerciements

Pendant ces trois ans, quel ne fut pas mon bonheur de retrouver la candeur enfantine qui sied à la vie d'étudiant. Loin de la classique stupeur d'amphi, je décidai au contraire de placer ma liberté retrouvée sous le joug d'un égo de chercheur en formation. Je remarquais très vite que les femmes et hommes qui m'entourèrent, semblaient tous mus de ludiques lubies relevant aussi bien d'une envie de *repousser les frontières de la connaissance* que participer à un grand *playdate* de 3 ans. Dans ces remerciements, je tiens à saluer tous les copains de jeu que j'ai pu fréquenter.

Tout d'abord je tiens à remercier mon directeur de thèse Daniel Négru qui m'a invité à jouer avec lui. Grace à ces excellentes explications sur la règle du jeu, il a pu me guider tout au long de la partie.

Je tiens également à remercier les membres de mon jury de thèse pour avoir arbitré ma soutenance avec rigueur et bienveillance. Leurs questions, idées et remarques ont été précieuses pour éviter de me faire attraper par l'Épervier.

Je veux exprimer ma gratitude aux grands des classes du dessus avec qui nous avons fait des devoirs en commun. Sans eux, je n'aurais jamais écrit autant de pages, ni rendu tout à temps. Merci donc à mes co-auteurs Yiping, Georges, Adlen, Pantelis, François, David, Panaiotis, Damien, Yacine et bien sûr Daniel.

Je souhaite également témoigner ma reconnaissance à mes potes de récré des bureaux 2^8 (F. J. S.) et $2^8 + 1$ (D. M. S.), ceux du mercredi après-midi de l'Afodib (R. K. T.), ceux du centre aéré du LaBRI (S. L. C.) avec qui on jouait au baby, ceux qui ont changé de bahut entre temps (M. A. D. V.), ceux que j'ai croisé en classe découverte à Dublin (T. S. B. C.), les élèves grecs, espagnols, italiens, allemands de la classe verte T-NOVA (G. P. M. V. B. J. A. A. M.), merci également à mon correspondant Adrien pour avoir participé à l'effort de relecture de la présente rédaction. Je salue aussi tous ceux que j'ai oubliés car ils n'étaient pas là le jour de la photo de classe.

Un énorme merci à Samahou, Florent, Débs et Aurore pour la préparation du somptueux goûter de 4h de thèse.

Ce travail n'aurait pas été possible sans le soutien de l'Université de Bordeaux et du LaBRI, qui m'ont donné l'argent de poche nécessaire à voir mes rêves en grand.

à Déborah et à toutes nos étoiles passées et futures

Contents

Remerciements iii

List of Figures ix

List of Tables xi

Acronyms xiii

1 Introduction 1
- 1.1 Thesis Contributions 2
- 1.2 Thesis Organization 3

2 Perspectives in Network Softwarization and Content Distribution 5
- 2.1 Motivations for Software Defined Networking 5
- 2.2 Motivations for Network Functions Virtualization 7
- 2.3 VNF-as-a-Service: the T-NOVA approach 9
- 2.4 Network Softwarization, an opportunity for Content Distribution challenges? 16
- 2.5 Conclusion 22

3 CDNaaS: Content Delivery Network as a VNF 25
- 3.1 Introduction 26
- 3.2 Background on existing CDN solutions 30
- 3.3 The CDNaaS proposal 35
- 3.4 Linux containers benefits for CDNaaS deployment 43

3.5 CDNaaS Integration and Validation 46
3.6 Conclusion and Future work 54

4 A Model for Content Delivery Collaboration: a VNF-as-a-Service Perspective 57
4.1 Introduction 57
4.2 CDNaaS in ISP network: which collaboration model? 60
4.3 A User-Centric Collaboration Model 75
4.4 Conclusion 88

5 Deployment and Optimization of Virtual Content Delivery Networks 89
5.1 Introduction 90
5.2 CDNaaS chain composition 93
5.3 CDNaaS Service Graph Embedding 100
5.4 CDNaaS Dynamic SLAs support 111
5.5 Conclusion 118

6 Conclusion and perspectives 121

A Résumé en Français 125
A.1 Motivation pour la virtualisation du CDN 125
A.2 CDNaaS: l'Implémentation de référence 126
A.3 Déploiement de CDNaaS dans le réseau opérateur 126
A.4 Sessions de contenue centrées sur l'utilisateur 127
A.5 Allocation de ressource pour les VNF 127
A.6 Conclusion 128

B List of publications 129

Bibliography 131

List of Figures

2.1 SDN Functional architecture 6
2.2 High level view of T-NOVA System Architecture 11
2.3 Cross-Chronology of technological mutations between Content Distribution Services, and ISP Internet access offers 17
2.4 Peak Period Traffic Composition 19

3.1 the 20 biggest Autonomous Systems per public peering bandwidth capabilities (Data exported from the PeeringDB API). 26
3.2 Trade offs for Content Delivery Networks Technologies. 27
3.3 High-level functional breakdown of a Content Delivery Network Service . . . 30
3.4 ETSI TISPAN functional architecture 31
3.5 Impact of DNS server on Airbnb server selection 32
3.6 Impact of DNS server on Youtube server selection 33
3.7 High Level Architecture of CDNaaS 37
3.8 Internal Architecture of VMG clusters 40
3.9 Comparison between T-NOVA and SwiftStack abstractions 41
3.10 Architecture of the Ingestion 42
3.11 Comparison between the Traditional VNF Lifecycle and the agile lifecycle . . 44
3.12 High Level architecture for a CDNaaS running in a NFV-as-a-Service platform 48
3.13 Sequence Diagram of CDNaaS, the vCDN Customer interacts with the T-NOVA Marketplace to create the service, and with the Element Manager for Service Configuration and monitoring. 49
3.14 Impact of the VMG HTTP filter/action overhead on content delivery 51
3.15 Impact of the load balancing on the VMG deployment 52
3.16 End-to-end evaluation of the content delivery components 53

3.17 Transcoding Performances 54

4.1 Deployement of a CDN Overlay using CDNaaS 61
4.2 CDN, ISP and Marketplace interactions 63
4.3 The vCDN Customer programs CDNaaS' CRO to orchestrate both internal resources (VMG and vSTR) and the vCDN Overlay Network Controller 64
4.4 Optimality zone for the CDN-as-a-VNF strategy (5) 68
4.5 Online CDNaaS management results. 72
4.6 Competition Loci in OTT content delivery. 75
4.7 Stakeholders interactions in the content session. 76
4.8 Blockchain-based model for collaborative video delivery. 78
4.9 CDN, CDNaaS and μCDN services deployed in an ISP network 82
4.10 Respective TEs share for CDC 85
4.11 Average price for content delivery 86
4.12 Testbed 86
4.13 Performance and scalability experiment 87

5.1 Canonical Model 95
5.2 Concrete Implementation of the CDNaaS Service using SFC 96
5.3 Building a tree with Service Edges. 97
5.4 2 Service-isomorphic graphs. 97
5.5 Phase I: Assigning vMG to CG 98
5.6 Phase II: Assigning vStreamer to vMG 98
5.7 Phase III : Partial Embedding without CDN, Assigning CDN to VMG 98
5.8 Mapped VCDN Service 100
5.9 Illustration of our mapping "genotypes", mutation and breeding procedures. 104
5.10 Cost Comparison of serie of successive embedding performed on the Geant Topology 108
5.11 Comparison of solvers computation time 109
5.12 Comparison of solvers performance 110
5.13 Adaptive strategy selection algorithm 110
5.14 VCDN five steps of deployment and operation 112
5.15 Step 1 – Data collection showing a typical mixture of ASs at a peering point . 112
5.16 Step 2 – Forecast, SLA and Discretization evaluations 114
5.17 Different discretization parameters 115
5.18 SLA generated from discretized predictions 115
5.19 Step 3 – Discretization and SLA Generation 115
5.20 Step 5 – Dynamic cost-aware scheduling 116
5.21 Example of the evolution of the cost of Service Embedding with SLA generated from 4 ISPs, 4 legacy CDN peering points over 24h 117
5.22 Evaluation impact of service optimization and discount policy 117

List of Tables

2.1 Main challenges faced by ISP and proposed solutions developed in this thesis . . . 21
2.2 SWOT Analysis of adopting the proposals (1), (2) and (3) of Table 2.1 . . . 22
2.3 List of major strategic opensource SDN and NFV projects or initiatives where Telcos assume director positions[a] . . . 23

3.1 Requirements for CDNaaS . . . 36
3.2 ETSI TISPAN and CDNaa+T-NOVA Functional Architecture Comparison . . . 38
3.3 REST API used to configure the VMG Filters and Actions . . . 39

4.1 Notations and estimates for use-case 1 . . . 65
4.2 Profitability matrix containing Earnings and Costs for CDN and ISP for each collaboration scenario . . . 66
4.3 Comparison of different CDN deployment models . . . 71

5.1 CDNaaS SLA description . . . 94
5.2 Notations . . . 101
5.3 # of Services graphs and computation time . . . 106
5.4 Cost Parameters . . . 106
5.5 Genetic Algorithm parameters . . . 108

Acronyms

ALTO Application-Layer Traffic Optimization.

API Application Programming Interface.

AS Autonomous System.

CAPEX Capital Expenditure.

CBC Content Brokering Contract.

CDC Content Distribution Contract.

CDCS Content Delivery Service Description.

CDN Content Delivery Network.

CDNaaS CDN-as-a-Service.

CDNi CDN Interconnection.

CLC Content Licensing Contract.

CP Content Provider.

CPE Customer Premises Equipment.

CPU Central Processing Unit.

CRO Caching and Routing Orchestrator.

DNS Domain Name Server.

DPDK Data Plane Development Kit.

DRM Digital Rights Management.

EMS Element Management System.

EU End-User.

HTTP Hypertext Transfer Protocol.

HTTPS HTTP Secure.

ILP Integer Linear Programming.

IOT Internet of Things.

ISP Internet Service Provider.

IVM Infrastructure Virtualization and Management.

IXP Internet Exchange Point.

NFV Network Function Virtualization.

NFV MANO NFV Management and Orchestration.

NFVI Network Function Virtualization Infrastructure.

NFVI-POP Network Function Virtualization Infrastructure Point of Presence.

NMS Network Management System.

NS Network Service.

NSD Network Service Descriptor.

NSP Network Service Provider.

OPEX Operational Expenditure.

OSPF Open Shortest Path First.

OTT Over-The-Top.

P2P Peer-to-peer.

PBFT Practical Byzantine Fault Tolerant.

POP Point-of-presence.

QoE Quality of Experience.

QoS Quality of Service.

SDN Software Defined Networking.

SFC Service Function Chaining.

SLA Service Level Agreement.

TE Technical Enablers.

TeNOR T-NOVA Orchestrator.

URL Universal Resource Locator.

vCDN Virtual CDN.

vCPU Virtual CPU.

VDU Virtual Deployment Unit.

VIM Virtual Infrastructure Manager.

VM Virtual Machine.

vMG Virtual Media Gateway.

VNE Virtual Network Embedding.

VNF Virtual Network Function.

VNFC Virtual Network Function Component.

VNFD Virtual Network Function Descriptor.

VNFM Virtual Network Function Manager.

VOD Video on Demand.

vStreamer Virtual Streamer.

WAN Wide Area Network.

WICM Wide Area Network Infrastructure Connection Management.

CHAPTER

1 Introduction

For many years now, the consumption of Over-The-Top (OTT) video-on-demand streaming services has been constantly increasing. In 2016, such services accounted for more than 55 percent of peak Internet traffic, and that percentage is still growing [SANDVINE, 2016], expecting to reach more than 80 percent in 2020. Shifts in content distribution have thus become necessary, leading to Content Delivery Networks (CDNs) and other approaches that accommodate OTT content.

The content distribution ecosystem that has emerged during the past decade is structured around a variety of actors along a value chain. Some actors fulfill a technical role, such as Internet Service Provider (ISP) and CDN operators, whereas others are more oriented toward business, such as Content Provider (CP).

This value chain has governed the democratization of online videos, but the current surge of OTT IP-based streaming reshuffles the deck. Specifically, ISPs are left behind, because they do not benefit from the added value of content delivery (even though they are still supporting heavy infrastructure costs), and CDN operators must build increasingly expensive networks to target a worldwide audience. At the same time, CPs and owners are benefiting from growth in the online videos market. In this ecosystem, collaboration between actors is the key to tackle the existing challenges, with a specific focus on finding synergies.

ISPs are solicited by CPs or CDNs to install streaming appliances within their network to deliver content directly to their users (e.g., Netflix Open Connect), reducing the number of hops as well as inter-autonomous-system traffic [Pathan, 2014a]. These solutions reduce both CDN and ISP costs and ease technical cooperation by enabling joint control of the delivery [Böttger et al., 2016]. However, the collaboration is not fair, because the ISP does not benefit from the revenue generated by the CP. Furthermore, many companies are now consolidating the different roles of the value chain into one entity [Hallingby et al., 2016].

Although this might facilitate end-to-end operations, it can hinder innovation and raise the barrier for newcomers on the market.

At the same time, existing technologies initially developed to support IT services such as cloud computing and network programmability are starting to emerge in the Telecom world as "*network softwarization*":

- Software Defined Networking (SDN) is the latest incarnation of a long history of efforts to make computer networks more programmable [Feamster et al., 2014], and paradoxically revisits ideas from early telecom networks related to the separation of control and data planes to simplify network management and the deployment of new services.

- Network Function Virtualization (NFV) was introduced in 2012 by the European Telecommunications Standards Institute through a seminal white paper [Virtualisation, 2012]. The NFV concept aims at creating a reference architecture and a standardized approach to achieve carrier grade virtualization on commodity servers of existing network functions that are currently handled by hardware middleboxes.

We believe that these emerging paradigms offer an interesting technological platform to address the challenges faced on the content delivery front.

Based on these observations, the virtualization of the content delivery function seems to be a promising approach to address the aforementioned issues. In this thesis, we study how a collaboration between ISPs, CPs and CDNs centered on a Virtual CDN (vCDN) can take place to improve End-Users Quality of Experience while reducing costs and fairly allocating profit along the content distribution value chain. Bellow, we outline the specific contributions made in this thesis.

1.1 Thesis Contributions

In the context of promoting collaboration for content delivery, our significant contributions are:

1. First, as current advances on network softwarization allow deploying Virtual Network Functions in an operator network, we propose the virtual CDN-as-a-Service (CDNaaS) concept. It allows a customer to deploy its own implementation of a CDN as a Virtual Network Function (VNF) at the edge of the ISP network. We present its architecture and design, as well as the integration into an open-source NFV Management and Orchestration (NFV MANO) stack developed within the T-NOVA European project. We evaluate each component according to a set of requirements and the system as a whole, at large scale. We also discuss and demonstrate how automation and scalability can be achieved easily with the use of Linux containers.

2. Second, as the benefits for a virtual CDN depends on its adoption by content delivery actors, we study how a mutually profitable collaboration can be established between ISP and CP/CDN operators and present the architecture and virtualization targets for a CDNaaS-based deployment. We advocate for specifying the service through a high-level Service Level Agreement (SLA) preserving the confidentiality of the ISP Network. We then analyze the current collaboration schemes and model them using game theory in order to find the conditions for which CDNaaS is optimal, considering a realistic pricing model.

3. In order to complement the proposed model, as the content delivery value chain is evolving with the rise of OTT services, we extend our proposal to the End-User (EU). We propose a user-centric model that considers each video viewing as a cooperative effort between several actors. To this end, a brokering mechanism is foreseen using the Blockchain and Smart Contracts to mix and match each stakeholder that can provide the adequate Quality of Experience (QoE) to the EU with the lowest price. The proposal is evaluated with a real implementation and a discussion on the scalability and governance is triggered. Our findings suggest that such a brokering mechanism is feasible, given that significant progresses on Blockchain scalability are made.

4. Finally, while the CDNaaS concept seems promising, deploying it on ISP network has practical implications. After recalling the main aspects of the general VNF resource allocation problem, we propose a solution for each sub-step. New algorithms are presented to (1) generate the Service Functions Chain corresponding to a vCDN SLA, (2) realize the embedding of a vCDN service on the ISP physical network and (3) schedule the service while supporting Dynamic SLAs. Our results indicate that to reduce the hosting cost of an SLA in its network, ISPs should put in place a discount pricing model to incentive its clients to generate longer lasting SLAs.

1.2 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 overviews the main ideas of network softwarization and suggests that it could be the next evolution of the technological mutation for ISPs to solve content distribution issues. Chapter 3 introduces the proposed concept of CDNaaS and highlights its deployment on top of an NFV platform [Rebahi et al., 2016] while Chapter 4 presents two collaboration schemes leveraging CDNaaS targeting cost reduction and QoE improvements [Herbaut et al., 2016] [Herbaut and Négru, ress]. In Chapter 5, we study the resource allocation problem for a virtual CDN [Herbaut et al., 2017b] [Herbaut et al., 2017a] before concluding and presenting pointers to directions of future work in Chapter 6.

CHAPTER

2 Perspectives in Network Softwarization and Content Distribution

Software Defined Networking (SDN) and Network Function Virtualization (NFV) are different expressions of the overall transformation trend toward *network softwarization*, which is deeply impacting and bridging the Telecom and IT industries [Freeman and Boutaba, 2016]. This new paradigm promotes software-controlled treatment of flows in the network and the orchestration of resources to meet the needs of customer applications in a converged network and cloud infrastructure.

In this chapter, we present an overview of network softwarization technologies and explain what the expected benefits are. We then detail a concrete implementation of a converged NFV platform that adds a new business perspective to the technology thanks to an innovative concept of Marketplace. We conclude this chapter by describing the evolution of content distribution technologies and discussing what are the opportunities brought by network softwarization to address to the outstanding challenges linked to the rise of on-line multimedia entertainment.

2.1 Motivations for Software Defined Networking

Today, distributed control and transport protocols are a key technology allowing information to travel around the world. Protocols such as OSPF and BGP are central to the configuration of their network by Internet Service Providers (ISPs).

They suffer however from several problems. They are notably hard to manage since ISPs need to express high-level policies in low-level and vendor specific fashion. Another difficult task is to support highly dynamic environments where traffic surges and faults are very common and need to be addressed in a timely manner. For that, the goal of having automatic reconfiguration and response mechanisms are very difficult in current IP networks. Moreover, the fact that networks are vertically integrated mixing data-plane

and control-plane functionalities in the same device makes innovations very difficult to roll-out. For example regarding IPv6, despite the exhaustion on IPv4 addresses and the fact that RFC2460 was published in 1998, adoption is still very low reaching less than 20% of all queries handled by Google Search engine[1].

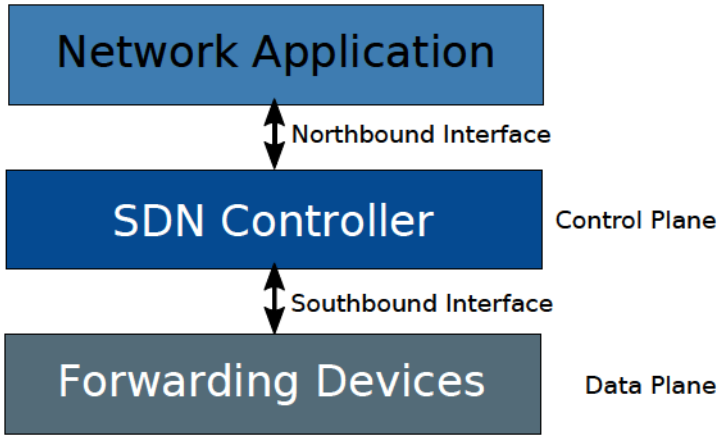


Figure 2.1: SDN Functional architecture

Drawing on these observations, the need for more flexible industrial network management technologies gained momentum. SDN is an emerging paradigm that gives hope in changing the above-mentioned limitations of the network. It breaks the vertical integration by separating the control logic (the control plane) and routers and switch that forward the traffic (the data plane) (Figure 2.1). In this context, Network switches become simple forwarding devices and the logic is implemented in a logically centralized controller. The SDN Controller is the central layer that is responsible for receiving (through its northbound interface) the high-level policies expressed as software in the Network Application and translate them to low-level forwarding rules deployed (through its southbound interface) on the Forwarding devices.

Even if the technological fragmentation the northbound interface is still an open issue and is subject to a fierce competition from SDN Controller projects (see Table 2.3), the southbound interface is standardized de-facto by the Open Network Foundation's OpenFlow [McKeown et al., 2008]. Supported by major devices vendor and software implementation (e.g. Open vSwitch [Pfaff et al., 2015]), it is an undeniable commercial success driving the adoption of SDN.

Even if the SDN principle can be subject to architectural interpretation, is can be reduced to four essential pillars [Kreutz et al., 2015]:

- control and data planes are **decoupled**

[1] https://www.google.com/intl/en/ipv6/statistics.html

- Forwarding decisions are **flow-based** instead of destination based, as all packets of a flow receive identical services policies at the forwarding device
- **Control logic** is moved to an external entity, the SDN Controller (also called the Network Operating System)
- the network is **programmable** through software applications running on the NOS

The expected benefits are simpler and less error prone network policies modification thanks to high-level language and software components compared to low-level device-specific configuration. Interestingly, a control program can react to spurious changes whether they come from change in the network state or from bad low-level rules deployed on the devices. Finally, SDN offers a global knowledge of the network state which simplifies the development and allows for creating more sophisticated networking functions and services.

Even if SDN scalability is sometimes subject to discussion, especially regarding controller scalability, flow initiation overhead, resiliency to failures, the lack of northbound standardization and network programming and management complexity [Yeganeh et al., 2013], those problems are not fundamentally specific to SDN and could be overcome without losing its benefits. Several industrial success stories following the adoption of SDN in a wide range of networking problems are documented in research from software defined WAN [Jain et al., 2013], to data-center networking [Singh et al., 2015] to recent deployment at the edge [Yap et al., 2017].

The programmability model of Openflow also suggests that SDN is not yet capable of handling every networking use-case. Despite recent solutions proposed to improve its processing capabilities such as P4 [Bosshart et al., 2014] and PMP [Pontarelli et al., 2017], it is not clear which middle box can be replaced by SDN-programmable hardware. If simple state-full network functions such as Firewall [Suh et al., 2014] can be implemented, more complex processing such as Video Transcoding and storage-intensive functions such as caching are out of reach for SDN. Indeed, forwarding devices should be function-agnostic to keep hardware prices down, and specialized processing may not be available. For this reason, NFV is perceived as the other promising paradigm of network softwarization, allowing network functions to be virtualized to replace middle boxes by software.

2.2 Motivations for Network Functions Virtualization

This section is a brief overview of NFV, which illustrates the main motivations behind this new service architecture [Mijumbi et al., 2016]. More details about the standard components including the Orchestration management and the Infrastructure Virtualization and Management (IVM) are covered in the presentation of a real implementation, the T-NOVA project, in Section 2.3.

Today, service provisioning in the ISP infrastructure relies heavily on the deployment of middleboxes, each of which performs a different Network Function. They are designed from costly specialized hardware to support the carrier-grade speed and stability required in the Telco world. Even if this model proved efficient in supporting their network until today, the current need to deploy new services quickly forced ISPs to reconsider it.

Middleboxes product cycles are very long and impede innovations from being quickly rolled-out. Middleboxes placement is a heavy procedure that requires fine-tuning and complex network configuration updates. For this reason, it participates to the so-called ossification of the Internet infrastructure [Council et al., 2001]. Given the fact that the number of middleboxes is roughly equivalent to the number of routers [Sherry et al., 2012], it represents a very important share in ISPs Capital Expenditure (CAPEX) and Operational Expenditure (OPEX).

The concept of NFV[2] was introduced in 2012 [ETSI, 2012] from a consortium of ISPs willing to specify the requirements for the deployment of Network Functions as software on Commercial off-the-shelf hardware. On top of the obvious goal of cost reduction, several interesting features are expected from this approach.

First, as a rule of thumb, since no manufacturing or hardware conception is involved, innovation happens faster through software. As a corollary, Virtual Network Function (VNF) providers collaborate easily on software through the adoption of open-source software development models [Hippel and Krogh, 2003]. VNF providers cooperate and mutualize their effort in solving common goals and reach better software quality, faster. For example in 2016[3] the open vSwitch project (an essential building block for both SDN and NFV) was transferred from VMWare to the Linux Foundation with large corporate contributors (Cisco, Ericsson, Huawei, HP, IBM, Intel, Red Hat and VMware). Thanks to its robust foundations, the project is used by vendors in their vNF implementation without requiring a complete rewrite of the middleware[4,5]. Second, the deployment of services is more flexible with NFV, as shared physical resources can be used to support different network functions. VNF can be instantiated in any NFV-capable environment and migrated depending on the need. Third, capacity provisioning can adapt dynamically depending on demand. VNFs can increase their performance by scaling-out and reduce their footprint through scaling-in.

The main concerns regarding NFV is its ability to live up to its promise of interoperability and easy resource management. This could be achieved in two distinct ways: (1) the main actors can collaborate on an open-source project until it becomes the de facto standard (e.g. the OPNFV project[6] promotes the integration on upstream projects into a common)

[2] http://www.etsi.org/technologies-clusters/technologies/nfv

[3] https://www.linuxfoundation.org/press-release/open-vswitch-joins-linux-foundation-open-networking-ecosystem/

[4] The same model can also be valid when considering software support for hardware enhancements like DPDK, which was initially developed by Intel but is now also managed as a Linux foundation project

[5] other examples exist, such as pfSense for security, opensips for session border controllers

[6] https://www.opnfv.org/

and (2) since ETSI goal with NFV is not to produce standards but requirements, liaison with Standards Developing Organizations are developed[7].

Another risk undermining the NFV adoption is the fear that software performance would not match line-rate requirements of ISP. To overcome these concern, several initiative aims at making hardware acceleration available to Virtual Machine (VM) through the Network Function Virtualization Infrastructure (NFVI) layer. SR-IOV [Dong et al., 2012] allows bypassing the virtualized NIC provided by the Hypervisor by using direct memory access. In this case, however, VM cannot be migrated, which reduces the flexibility. Using FPGA SoC-based Compute Nodes[Karras et al., 2016] or GPU-enhanced [Paglierani, 2015] node can also improve performances, at the price of an high specialization of the server[8]. A pragmatic approach combining physical functions when required and VNF when possible is a viable option while performance is still an issue on today's server. This technique was used by on the first successful field study of virtual Home Gateways [Cantó Palancar et al., 2015], [Proença et al., 2017].

Thanks to the increased network flexibility induced by SDN and the possibility to deploy virtual appliances through NFV, they are perceived as key enablers allowing ISPs to upgrade their infrastructure to support new services, especially related to content distribution. In the next section, we dig deeper into the integration of NFV and SDN by describing the architecture of a real NFV platform, T-NOVA. Since T-NOVA is MANO-compliant, most of the considerations present in the following section apply to other NFV implementations, listed in Table 2.3.

2.3 VNF-as-a-Service: the T-NOVA approach

In this section, we briefly describe all the relevant aspects of the T-NOVA project [Xilouris et al., 2015] that were used to deploy and evaluate our contributions. The content of this section was adapted from cited T-NOVA deliverables and cited partner papers.

T-NOVA, "Network Functions as-a-Service over Virtualized Infrastructures" is a European FP7 Large-scale Integrated Project, whose primary aim is the design and implementation of a management/orchestration framework for the automated provision, configuration, monitoring and optimization of Network Functions-as-a-Service (NFaaS) over virtualized Network and IT infrastructures. T-NOVA leverages and enhances cloud management architectures for the elastic provision and re-allocation of IT resources hosting Network Functions. It also exploits and extends SDN platforms for efficient management of the network infrastructure.

The T-NOVA framework allows operators to deploy virtualized network functions, not only for their own needs, but also to offer them to their customers, as value-added services.

[7] https://portal.etsi.org/tbsitemap/nfv/nfvliaisonmatrix.aspx

[8] this solution has also be recently adopted by cloud providers: https://aws.amazon.com/ec2/elastic-gpus/

Virtual network appliances (gateways, proxies, firewalls, transcoders, analyzers etc.) can be provided on-demand "as-a-Service", eliminating the need to acquire, install and maintain specialized hardware at customer premises.

Leveraging this NFaaS concept and in order to facilitate the involvement of diverse actors in the Network Function Virtualization scene as well as the attraction of new market entrants, T-NOVA introduces a novel concept of "NFV Marketplace", in which network services and Functions offered by several developers can be published and brokered/traded. The NFV Marketplace enables customers to browse and select services and virtual appliances that best match their needs, as well as negotiate Service Level Agreements (SLAs) and be charged under various billing models. A novel business case for NFV is thus introduced and promoted.

2.3.1 Overall Architecture

The T-NOVA system architecture [Kourtis et al., 2017] inherits the majority of its concepts from the generic ETSI NFV ISG architectural model [ETSI, 2013a] and expands it with specific add-on features. The T-NOVA architecture encompasses four key architectural layers (as shown in Figure 2.2):

- The NFVI layer consists of both physical and virtual nodes (high-volume servers, Virtual Machines, storage systems, switches, routers etc.) on which the network services are deployed; The T-NOVA platform was deployed through 3 Network Function Virtualization Infrastructure Point of Presences (NFVI-POPs) in Greece, Portugal and Germany.

- The NFVI Management layer includes the infrastructure management entities: Virtual Infrastructure Manager (VIM) and the Wide Area Network Infrastructure Connection Management (WICM). T-NOVA adopts an OpenStack[9] cloud operating system for control of the compute and data-center assets and OpenDaylight[10] for the control of the network infrastructure (most of which is SDN -based);

- The Orchestration layer is based on the T-NOVA T-NOVA Orchestrator (TeNOR) Orchestrator [Riera et al., 2016] and also includes a "Network Function Store" which is a repository for all published VNFs. The Orchestrator, along with the NFVI Management layers comprise the T-NOVA NFV Management and Orchestration (NFV MANO) stack;

- Finally, the Marketplace layer contains all the customer-facing interfaces and modules, which facilitate multi-role involvement and implement business-related functionalities.

[9] https://www.openstack.org
[10] https://www.opendaylight.org/

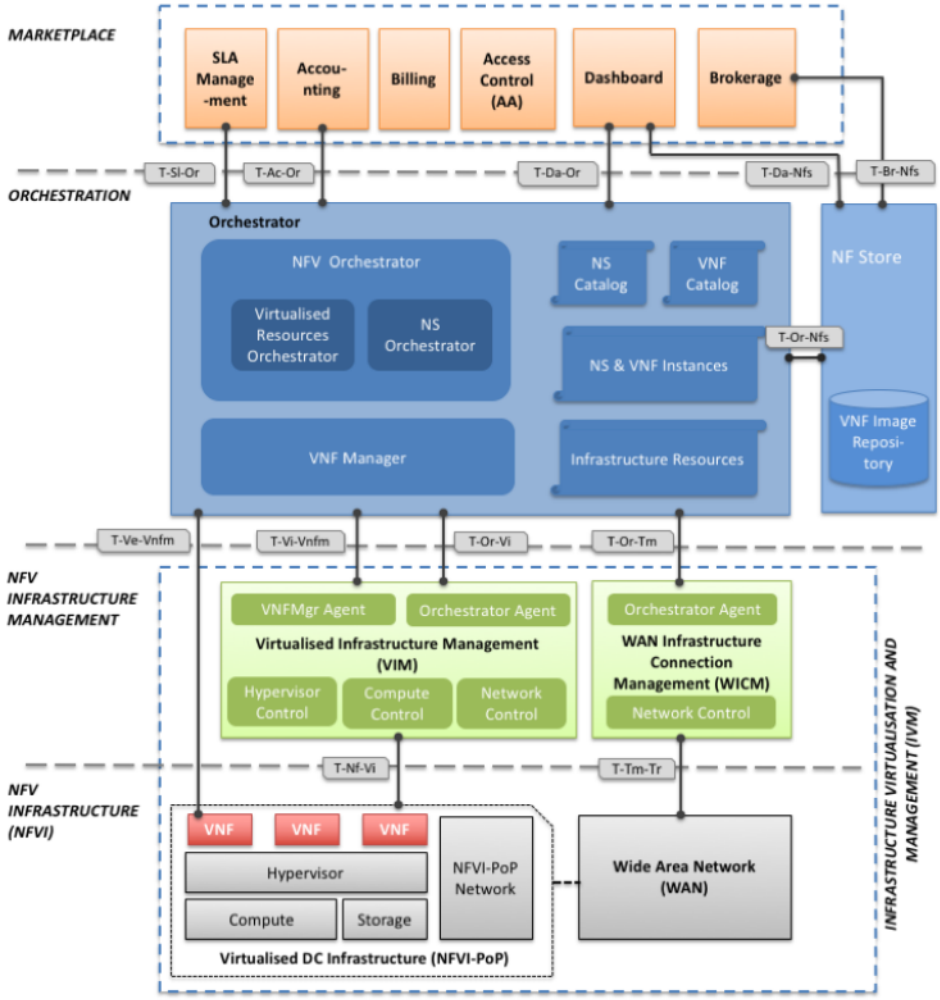


Figure 2.2: High level view of T-NOVA System Architecture

The three following sections present the functional overview and architecture details on each of these layers.

2.3.2 NFVI Management Layer

T-NOVA's Infrastructure IVM layer [T-NOVA Consortium, 2016a] provides the required hosting and execution environment for VNFs. The overall design of the layer is driven by a variety of requirements such as performance, elasticity etc.

2.3.2.1 Functional Overview

The IVM incorporates a number of key concepts that influence the associated requirements and architecture for the layer:

- The IVM supports separation between control and data planes and network programmability. The T-NOVA architecture leverages SDN for designing, dimensioning and optimizing control- and data-plane operations separately, allowing capabilities from the underlying hardware to be exposed independently.
- The IVM is based around the use of clusters of commodity computing nodes in cloud computing configurations to support instantiation of software components in the form of VMs for NFV support, offering resource isolation, optimization and elasticity.
- The IVM exposes the necessary interfaces to support appropriate integration. The external interfaces provide connectivity with the T-NOVA Orchestration layer while the internal interfaces provide connectivity between the internal domains of the IVM to ensure the requests for the creation, deployment, management and termination of VNF services and their host VMs can be executed

2.3.2.2 Architecture

The VIM manages both the IT (compute and hypervisor domains) and network resources by controlling the abstractions provided by the Hypervisor and Infrastructure network domains. It also implements mechanisms to efficiently utilize the available hardware resources in order to meet the SLAs of Network Services (NSs). The VIM is also plays a role in the VNF lifecycle management. Additionally, it collects infrastructure utilization/performance data and to make this data available to the Orchestrator in order to generate usage/performance statistics, as well as triggering scaling.

It is composed of the following modules:

- The **WAN Infrastructure Connection Manager** provides the link between WAN connectivity services and the NFVI hosting VNFs including connectivity to NSs allocated in more than one NFVI-POP.
- The **VIM Compute Control** provides an appropriated performance level for VNFs.
- The **VIM Hypervisor Control** implements hardware resource abstraction, virtual resource lifecycle management mechanisms.
- The **VIM Network Control** implements an SDN approach to provide network virtualization capabilities inside a NFVI-POP. It supports transport tunneling protocols of L2 packets over L3 networks, to assist the WICM in setting up the communication/ between different NFVI-POPs.

Each module is responsible for collecting metrics and make them available to the orchestrator layer through the appropriate agents.

2.3.3 Orchestration Layer

An Orchestrator Platform is a central technology component in enablement of Network Function Virtualization and Software Defined Networks in carrier grade networks. The Orchestrator plays a key role in enabling performance, scalability, availability and openness. It is the Orchestrator's role to map new services' requests onto the existing infrastructure in an automatic, secure and efficient way, without ever being a business or operational bottleneck. T-NOVA's orchestrator, named TeNOR[11], is aligned with ETSI architecture requirements [ETSI, Network Functions VirtualisationV., 2014]. Other alternative orchestrator implementation were considered[12], such as Open Baton[13], OSM[14] and ONAP[15].

2.3.3.1 Functional Overview

Due to its pivotal role in the T-NOVA architecture, the Orchestrator implements appropriate interfaces to manage the interaction with the layers above and below it. Specifically, the Orchestrator provides:

1. A Northbound interface to the Marketplace and the Network Function Store;
2. A Southbound interface to the VIM. This interface supports the exchange of metrics data generated both at the infrastructure level and at the VNF /NS level. These metrics have to be collected (and transposed) and communicated to the Orchestrator in order for the Orchestrator to identify and inform the VIM what actions are required to be taken so that the NS SLA is maintained.

2.3.3.2 Architecture

TeNOR 's modules are briefly described in the following:

- **NS/VNF Manager**: it is a facade for the northbound interface (the Marketplace for the NS Manager, the NS Manager for the Virtual Network Function Manager (VNFM)), and manages the NS/VNF Catalog. The proposed architecture embraces both the concept of generic VNFM as well as VNF specific VNFMs, as suggested by ETSI [ETSI, Network Functions VirtualisationV., 2014].

[11] https://github.com/T-NOVA/TeNOR

[12] At the time T-NOVA project was started, these initiatives were not mature enough to be integrated. As of writing, significant progress has been made and forking an existing project would be advisable

[13] https://openbaton.github.io

[14] https://osm.etsi.org/

[15] https://www.onap.org/

- **Service Mapping**: this module contains the mapping algorithm implementations, which map the required resources to support a NS instance to the best available location in the infrastructure respecting the constraints posed by: (i) the current availability of network infrastructure resources, (ii) the type and amount of resources demanded by the services to be mapped and (iii) SLA specific needs. Several Service mapping algorithms have been proposed in T-NOVA based on Integer Linear Programming (ILP) formulations and Reinforcement learning [Liberati et al., 2017]. The approach adopted to maximize the infrastructure provider revenue is trying embed as many service mapping requests as possible.

- **NS/VNF Provisioning**: it accepts requests for NS instances from the Marketplace (through the NS Manager) and for VNF instances from the VNFM ; it also manages the NS/VNF Instances repositories

- **NS/VNF Monitoring**: it accepts Virtual Machine based monitoring data from the lower VIM layer and maps it to the corresponding NS/VNF instances. This data is later given to the Marketplace, for both Customers and Function Provider dashboards;

- **SLA Enforcement**: responsible for comparing monitoring data to the agreed SLA for every NS instance, and generates alerts for impending SLA breaches. Data associated with a potential breach is passed to the NS Manager, which initiates the necessary actions to guarantee the SLA (it either migrates or scales VNF instances or improves their network connections)

- **Resource Repository**: provides infrastructure related information collected from the VIM and NFVI components of the Infrastructure and Virtualization Management

- **IVM Layer**: The IVM layer in the T-NOVA system is responsible for providing the execution environment for VNFs. The IVM is comprised of a number of domains including the NFVI and the VIM and WICM. The IVM provides full abstraction of the NFVI resources to VNFs.

- **WICM**: The WICM is responsible for providing the link between Wide Area Network (WAN) connectivity services and the NFVI hosting VNFs including connectivity to NSs allocated in more than one NFVI-POP. The WICM has a dedicated TeNOR interface to receive requests for allocating WAN connections to services.

2.3.4 Marketplace Layer

T-NOVA introduces the concept of a Marketplace in an NFV framework [Xilouris et al., 2014]. The aim of the Marketplace is to promote VNF service offerings and facilitating commercial activity and seamless interaction among the various business stake-holders interacting with the T-NOVA system.

2.3.4.1 Functional Overview

The T-NOVA Marketplace provides an intuitive interface to the underlying NFV MANO stack. Four are the main functionalities identified:

- *Publication of resources and NF advertisement.* Through a customer front-end, Third-party VNF developers describe their functions in the T-NOVA Function Store, and customers place their requests for services and virtual appliances.
- *VNF discovery, resource trading and service matching.* Through a brokerage module customers can place their requests for T-NOVA services and declare their requirements for the corresponding VNFs, receive offerings and make the appropriate selections, taking into account the offered SLAs [Markakis et al., 2016]. Trading and billing policies such as long-term lease, scheduled lease, short-term lease or spot markets can be based either on a fixed-price or action-based strategies.
- *Customer-side monitoring and configuration of the offered services and functions.* Via a service dashboard users can interact with the T-NOVA Orchestrator platform for monitoring the status of the established services and associated NFs, as well as for performing — according to their associated permissions — management operations on them [Gardikis et al., 2016].
- *Billing of services.* This includes the establishment of pricing mechanisms for different NF and the study of how these prices are affected by SLAs evaluation, so that customers may receive certain compensations depending on the overall service delivered [Skoviera et al., 2017].

2.3.4.2 Architecture

The Marketplace is composed of various micro-services (see Figure 2.2) with the following roles:

- The **Business Service Catalog** stores all the available service offerings in the marketplace.
- The **Brokerage Module** computes for a given Network Service SLA, each type of VNF in the Service Functions Chain, which VNF provider offers the most cost-effective alternative.
- The **Dashboard** provides the Graphical User Interface for all customer-facing services. It can be used by function providers, service providers and customers.
- The SLA Management Module establishes and stores the SLAs among all the involved parties and checks if they have been fulfilled or not . It informs the accounting system for the pertinent billable items (penalties or rewarding).

- The **Accounting Module** stores all the information needed for later billing: usage resources for the different services and SLAs evaluations
- The **Billing Module** produces the bills based on the information stored in the accounting module.

2.3.5 Summary

The functional specifications of the Orchestrator and IVM are tightly adapted from ETSI requirements, and are common to all NFV MANO-compliant implementation of NFV. As a consequence, having a VNF deploys on T-NOVA assures a good portability to other platforms.

By integrating an innovative marketplace layer, the business aspects are taken into account. SLAs allow high-level network services to be deployed on the platform with the assurance that the most cost-effective VNFs are selected in the service chain thanks to brokerage module.

Now that we have presented the most salient features of network softwarization, we discuss the opportunities in using them to solve Content Distribution issues in the operator network.

2.4 Network Softwarization, an opportunity for Content Distribution challenges?

In this section, we present the current state of the ISPs market, and the challenges they have been confronted throughout the evolution of content distribution business models and Internet Access since the inception of commercial Internet at the beginning of the 1990s.

We advocate that the new trend of Network Softwarization is the next technological evolution for Future Internet that will help ISPs staying profitable in an era where the commoditization of IP access and Over-The-Top (OTT) content distribution is prevalent.

For this, we first go through a little bit of history and compare the evolving roles of Content Provider (CP) and ISP in the content distribution value chain. Then we describe the virtualization target that we analyzed and developed during our research that constitute the next step toward a virtualized content distribution model. We finally present the challenges of the approaches and the lessons learnt.

2.4.1 Content Distribution Evolutions as a Drivers of technological mutations for Internet Access

In this section we detail the 3 main mutations of content distribution starting from the beginning of the commercial Internet age in the mid-1990 to the new developments seen

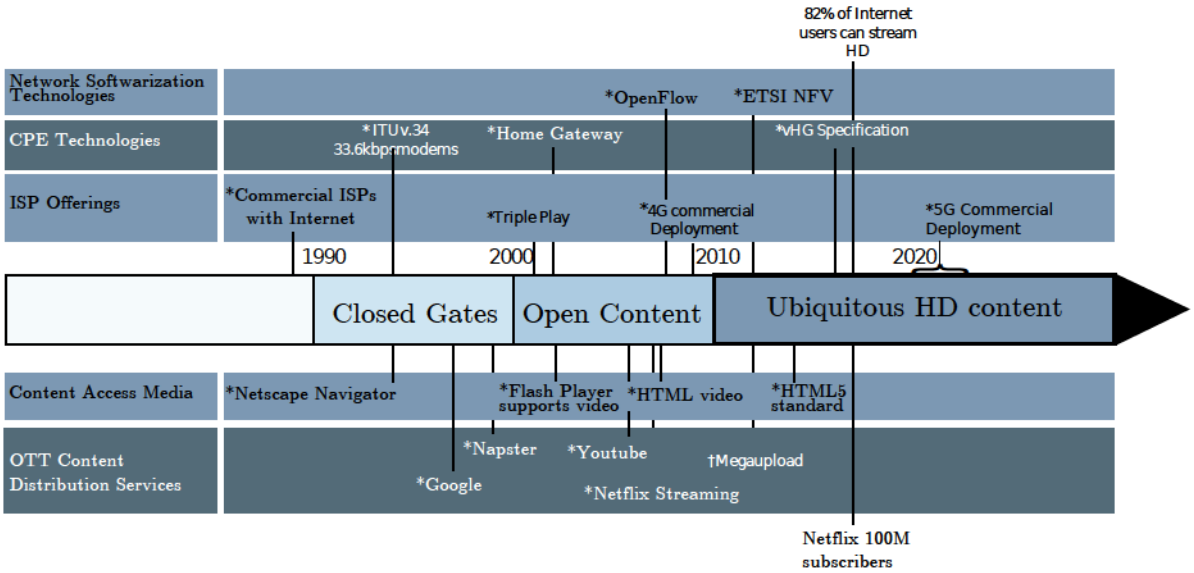
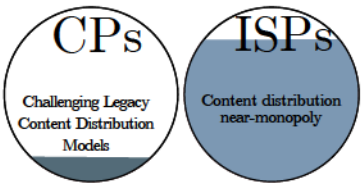


Figure 2.3: Cross-Chronology of technological mutations between Content Distribution Services, and ISP Internet access offers

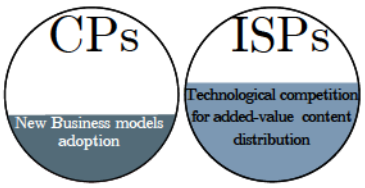
today. We advocate that the way users access their content on-line is a driver for change in the design and features for Internet access technologies. We conclude by stating that the current trend of over-the-top content access calls once more for innovation in the ISPs architectures.

2.4.1.1 1990-2000 Closed gates era: ISP Web portals and dial-up access



When Internet access started spreading in households in the mid-1990, End-Users (EUs) were receiving their Internet Connection using dial-up modems (v.34 or v.90/92). It was not uncommon for them to have only a limited access to the open Internet, through Service Providers Web portals such as AOL or Compuserve. Those now extinct web portals where paid for by the Internet subscriptions and were considered a safe haven since they only provided access to contents curated by the ISP for the EUs and licensed to them from traditional media such as printed newspapers, radio and television. While serving captive users from their web portals through their proprietary client software (such as CIM or America Online for Windows), ISPs had a very comfortable market positioning. This short-lived model was soon made obsolete due to EUs demanding faster connection speeds through xDSL technologies and the rise of gratis contents accessible from gratis web browsers with full HTML standard support such as Netscape Navigator.

2.4.1.2 2000-2010 Open Content era: Gratis Contents and Broadband access



At the turn of the XXth century, EUs started asking for services that were not provided through portals but instead made directly accessible on the open Internet. Internet Startups such as Google and Facebook invented new business models for contents and services, relying on smart content monetization through advertising, user targeting or affiliation [Evans, 2009]. Users were seduced by the promises of gratis contents, and developed a huge appetite for interactive and multimedia contents, driving up demand for broadband Internet. The Adobe Flash Player has been a key enabler that allowed users to access multimedia content regardless of their OS or browser and democratized video streaming since 2002.

This new content diffusion paradigm was specially disruptive for traditional written news media [Saba, 2009], which faced a sharp decline due to the advertising revenues moving away from physical medium to its on-line counterpart. TV and film industry quickly adapted to this new era by proposing Video on Demand services with initially only limited success, for two reasons. These services were competing against other form of gratis content distribution such as illegal peer-to-peer file sharing (such as Napster, which allowed users to share files between them without requiring a central server to host the files), illegal on-line hosting services (such as Megaupload, which allowed users to stream videos without having to download it in the first place). Another factor that temporarily hindered the shift towards paying contents is quality of experience. The broadband penetration was not sufficient to reach a critical customer mass, and the limited availability of downstream bandwidth and lack of proper content delivery networks, imposed subpar quality of experience wrt. premium TV subscriptions through IP/TV (offered through dedicated Set-top-boxes), proposed by ISPs in their new triple-play offers including Internet, telephone and TV access [Kelly et al., 2012].

To propose these new bundles and adapt to EUs requirements, Service Providers had to innovate and replace their broadband modems by more complex appliances to become proper Home Gateways[Holliday, 1997]. These always-on devices are designed to be reliable, remotely manageable and affordable. They are also extensible in the sense that they can support new physical peripherals and networks, as well as new services for security and multimedia [Den Hartog et al., 2004]. Standardization bodies also started issuing specifications for those complex devices [Broadband Forum, 2005], [Broadband Forum, 2006].

This era came to an end due to the conjunction of two different phenomenon: high-speed fixed and mobile Internet democratization and the fight against piracy.

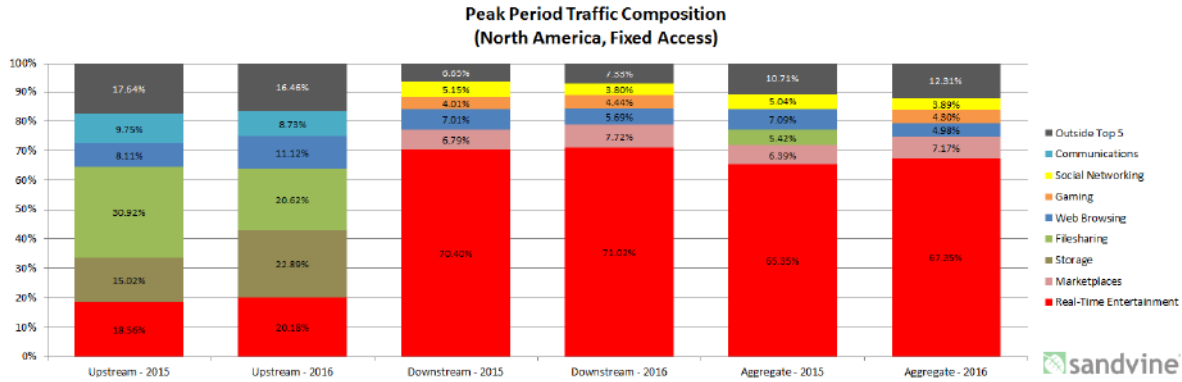
The years 2010 witnessed a dramatic increase in available bandwidth worldwide. In 2008 Globally only 20% of users had a +5Mbps Average Peak Connection Speed (with a mere 26% in the US) [Akamai Technologies, 2008]. In 2017 however, this ratio has risen to 82% globally (and virtually everywhere in the US, where half of the population can download at +15Mbps) [Akamai Technologies, 2017]. Mobile access also improved by 1

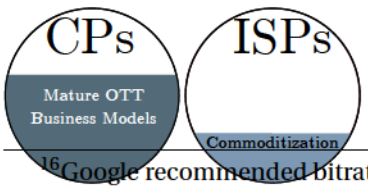
order of magnitude, providing on average 10Mbps in the US and often much more in Europe[SANDVINE, 2016]. With just 5Mbps bandwidth available, users can enjoy real-time video streaming with qualities ranging from standard 480p resolution to HD 720p [16], without the burden of waiting the download to complete, like in traditional peer-to-peer filesharing services.

This convenience was not the only reason why peer-to-peer have lost momentum. Following unprecedented lobbying [Hugenholtz et al., 2000] from Content Owners to protect their copyrights, legislators responded through the adoption of the WIPO Copyright Treaty [Ficsor, 2001] becoming effective in 2002, followed by its implementations in national or supranational legislation, such as the Digital Millennium Copyright Act [Lunney Jr, 2001] in the US and the Copyright Directive (Directive 2001/29/EC) in Europe [Westkamp, 2007]. As Pirate websites were shut down, users turned their back to peer-to-peer file sharing (from 60% of Internet bandwidth in 2006 [Ferguson, 2006] down to 3% today [SANDVINE, 2016]) to adopt gratis or paying Over-the-top providers as their main source of on-line entertainment.

2.4.1.3 2010+ Ubiquitous Streaming era: Multichannel distribution and Commoditized ubiquitous access

Figure 2.4: Peak Period Traffic Composition - North America, Fixed Access [SANDVINE, 2016]





Today, Over-the-top content distribution, and especially video content, has become prevalent. Combined with music streaming, it reaches 71% of Peak Period Downstream Traffic in North America as shown on Figure 2.4. To target a mainstream

[16] Google recommended bitrate https://support.google.com/youtube/answer/1722171?hl=en

market, OTT CPs adapted to EUs demands of seamless and ubiquitous access to streaming content by rolling out plethora of different content access technologies:

- dedicated websites accessible through web browsers [17], [18].
- third party applications downloaded on customers smart devices such as smartphones, tablets[19], smart TVs [Lee et al., 2013], or gaming console [20]
- Customer-purchased connected Digital Media Players (e.g. Apple TV, Amazon Fire TV, Google Chromecast)

With this vast choice of technology, EU can enjoy their OTT content wherever they have an IP connectivity. As ISP becomes more or less interchangeable and can only compete on prices, Internet connectivity is a commoditized market [Feamster et al., 2007]. As a result, Internet broadband access prices are decreasing over the years. For example, from February 2012 to autumn 2015, prices for broadband in the EU28 have fallen by 7.7% to 25.9% this decrease affects all types of service offerings and all speed basket across Europe [European Commission, 2016].

In this context, ISPs face three challenges presented in Table 2.1, for which we propose solutions detailed in the next section.

2.4.2 Network Softwarization, an opportunity for content delivery?

Fully aware of their difficult position, ISPs started embracing Network Softwarization, if not by deploying it into their network, then at least by heavily investing in its underlying technologies (NFV, SDN, Orchestration, Cloud platforms). On top of the traditional Standardization Bodies and Industry consortia which are considering the different challenges inherent to Network Softwarization (e.g. IETF RFC 7665 for Service Function chaining), Telcos are also contributing to the governance and workforce of open-source projects used as NFV and SDN building blocks as we reported in Table 2.3. What this trend is revealing, is that Network Softwarization is the next technological evolution chosen by the ISPs in order to address Challenge (1).

In this thesis, we decided to focus on specific issues related to the evolution of content distribution to the dawn of network softwarization. For this reason, we present a solution based on Network Softwarization that address challenges (2) and (3) through the deployment

[17] https://www.netflix.com/

[18] https://www.primevideo.com/

[19] 2011 https://media.netflix.com/en/press-releases/netflix-expands-support-for-android-powered-smartphones-and-tablets-migration-1

[20] https://media.netflix.com/en/press-releases/coming-soon-netflix-members-can-instantly-watch-movies-and-tv-episodes-streamed-to-tvs-via-the-playstation-computer-entertainment-system-migration-1

of a Virtual Content Delivery Network as Virtual Network Function sold through a VNF -as-a-service platform.

To further clarify what are the pros and cons of proposals (2) and (3), we conducted a SWOT analysis where we analyzed their strengths, weaknesses, opportunities, and threats. Table 2.2 summarize the main ideas.

Strengths: ISPs have invested a lot to build their robust high-speed physical infrastructures from IP-backbone to metro network to the last mile segment. These infrastructures can be leveraged to propose new services at scale. As EUs expectations in term quality of experience are very high, proposing a guaranteed bandwidth and delay is a key strength. ISPs are today the only actor capable of managing the network end-to-end, hence providing guaranties from the content server down to the user premises.

Weaknesses: Network Softwarization may require deploying new compatible hardware (forwarding devices supporting programmability, Network Function Virtualization Infrastructure Points of Presence). It may be hard for ISPs to invest in new technologies before existing ones reach their end-of-life in a commoditized market where the profit margins are thin. Another aspect that may impede the adoption of Network Softwarization is that extra effort is needed to adapt the internal organization of ISPs to accept and benefit from the technologies [Viginier, 2017] [Cantó Palancar et al., 2015].

Opportunities: Several opportunities can be expected by adopting Network Softwarization. First, running network functions as software can lower both CAPEX(capital expenditure) and OPEX (operational expenditure) especially in Deployment (Roll-out) and Upgrade, Capacity Management, Transport Network Operations, Service Assurance and Environmental Costs [Hernandez-Valencia et al., 2015]. Another benefit from Network Softwarization is the reduced time-to-market needed in order to roll-out new services [Han et al., 2015]. By design, deploying Virtual Network Functions only involves implementing features as

Table 2.1: Main challenges faced by ISP and proposed solutions developed in this thesis

	Challenges Descriptions	**Proposals**
Challenge 1	Reduce their costs to stay competitive in a commoditized Market	Adopt Network Softwarization by implementing NFV-capable Point-of-presences (POPs) and deploy VNFs (Chapter 3)
Challenge 2	Benefit from the OTT content delivery trend	Establish a collaboration with other actors of the content delivery value chain(Chapter 4)
Challenge 3	Increase their portfolio to sell new added-value services	Market added value services based on their infrastructure(Chapters 5)

Table 2.2: SWOT Analysis of adopting the proposals (1), (2) and (3) of Table 2.1

	Helpful	Harmful
Internal	**Strengths** End-to-end network management Existing infrastructure	**Weaknesses** New corporate Organization Investments in a Commoditized Market Technical Debt (legacy hardware)
External	**Opportunities** Lower CAPEXand OPEX Repurposable virtual resources Virtual CDN	**Threats** Deployment challenge in ISP Networks Must design collaboration models with CP

software, relegating the hardware design to commodity server vendors and leveraging open-source solution to implement the virtual infrastructure. Thanks to the support of multi-tenancy by SDN controllers [Muñoz et al., 2015], selling connectivity as a service is feasible [Feamster et al., 2007], [Manthena et al., 2015], [Aflatoonian et al., 2015]. Consequently, ISPs can create new business opportunities by implementing virtual overlays on top of their network or easily slicing them to sell premium network connectivity to third parties. Finally, Network Softwarization can be used to roll-out new access architectures that supports the fast deployment of new services for content delivery (e.g. Virtual Content Delivery Networks, see section 3.2). Deployed computing resources supporting Network Softwarization can be easily re-purposed to support new services.

Threats: Even if the perceived benefits are clear and migrating to a fully programmable network fabric has proven feasible and beneficial for Data Center Operators [Singh et al., 2015], it remains an open research topic for ISPs networks. Indeed, deploying softwarized ISPs networks have some practical implications such as current legacy hardware deployment upgrade [Poularakis et al., 2017], traffic heterogeneity, number of flows. It may be also problematic to maintain a global view of networks of such scale given the current architecture of control software [Rodriguez-Natal et al., 2017]. From a business perspective, ISPs need to assure that they can offer useful service for CPs. Designing a collaboration model between ISP and CP leveraging on the existing Strengths of ISPs may prove difficult (see section 4.2.2).

2.5 Conclusion

In this Chapter, we have provided an overview of network softwarization technologies and the possible opportunities they present to address content distribution challenge in the operator network. We believe these emerging paradigms are an opportunity to put the ISPs back in the game through cost reduction and new revenue stream through the deployment of novel content delivery solutions based on NFV. To support this idea we describe a model of a Content Delivery Network implemented as a VNF in the next chapter.

Table 2.3: List of major strategic opensource SDN and NFV projects or initiatives where Telcos assume director positions[a]

[a] Telcos are also involved in enabling technologies projects, such as Openstack, Data Plane Development Kit (DPDK)

	Initiatives	Missions	Telco acting as directors
Primarily SDN projects	OpenDayLight — openday-light.org	SDN Controller with Legacy equipment support, Promotes NFV	AT&T
	ONOS — onosproject.org	SDN Controller for ISPs	AT&T, China Unicom, Comcast, NTT, SK Telecom, Verizon
	CORD — opencord.org	To Bring Cloud Economies and Agility to the Telco Central Office (through SDN – ONOS- and NFV – OpenStack, XOS)	AT&T, China Unicom, Comcast, NTT, SK Telecom, Verizon
	ONF — opennetworking.org	Drive the OpenFlow Standard, accelerating the adoption of SDN & NFV	AT&T, Verizon, China Unicom, Comcast, NTT
Primarily NFV projects	OPNFV — www.opnfv.org	creates a reference NFV platform	Orange, China Mobile, Telecom Italia, Vodafone
	Open Mano — osm.etsi.org	develop an Open Source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV.	Telefónica, BT, Telenor
Telco Alliances and foums on NFV	ETSI, Metro Ethernet Forum, Alliance for Telecommunications Industry Solutions		
Telco Alliances and foums on SDN	ITU-T, Broadband Forum, Metro Ethernet Forum, Optical Internetworking Forum, Alliance for Telecommunications Industry Solutions		

CHAPTER

3 CDNaaS: Content Delivery Network as a VNF

As outlined in the previous chapter, Internet Service Providers (ISPs) start seizing the benefits of Network Softwarization by investing in Network Function Virtualization (NFV), Software Defined Networking (SDN) and next generation Points of Presence. However, to reap the benefits of Virtualization, porting existing cloud services is not enough:

- Cloud services are often coupled with the cloud platform they run on, as vendors do not use the same resource semantics, middleware technologies or interfaces [Silva et al., 2013]. This is unacceptable in the NFV world, as it promotes re-usability, portability and manageability as its core values.
- Telecom Operators need very reliable resources with *five nines* availability, since service availability for ISPs is often a regulatory requirement, as telecommunication networks are considered to be part of critical national infrastructure [ETSI, 2015]. Cloud providers are subject to frequent outages, for example on Feb. 28th 2017, the Amazon Storage S3 service faced a 2 hours outage on its Northern Virginia Region caused by a human error, dropping the uptime of the service to 99.977%.
- Virtual Network Function (VNF) management must also comply with existing Telecom Operator standards (e.g. UIT-T series X.700) and de-facto standards proposed by industry fora, which are not implemented by Cloud providers.

In this context, implementing Content Delivery over NFV has practical implications that go far beyond simply porting existing Content Delivery Network (CDN) solutions. In this chapter, we present CDN-as-a-Service (CDNaaS): our concept of a virtualized solution for content delivery over a NFV infrastructure. The motivations, architecture, interface integrations, design choices, scalability and performance evaluation of CDNaaS are exposed in detail. This first step toward the virtualization of the content delivery process demonstrates

the feasibility of deploying such service over a NFV infrastructure with enough flexibility and performances to support advanced use-cases for ISPs, Content Providers (CPs) and CDN providers.

3.1 Introduction

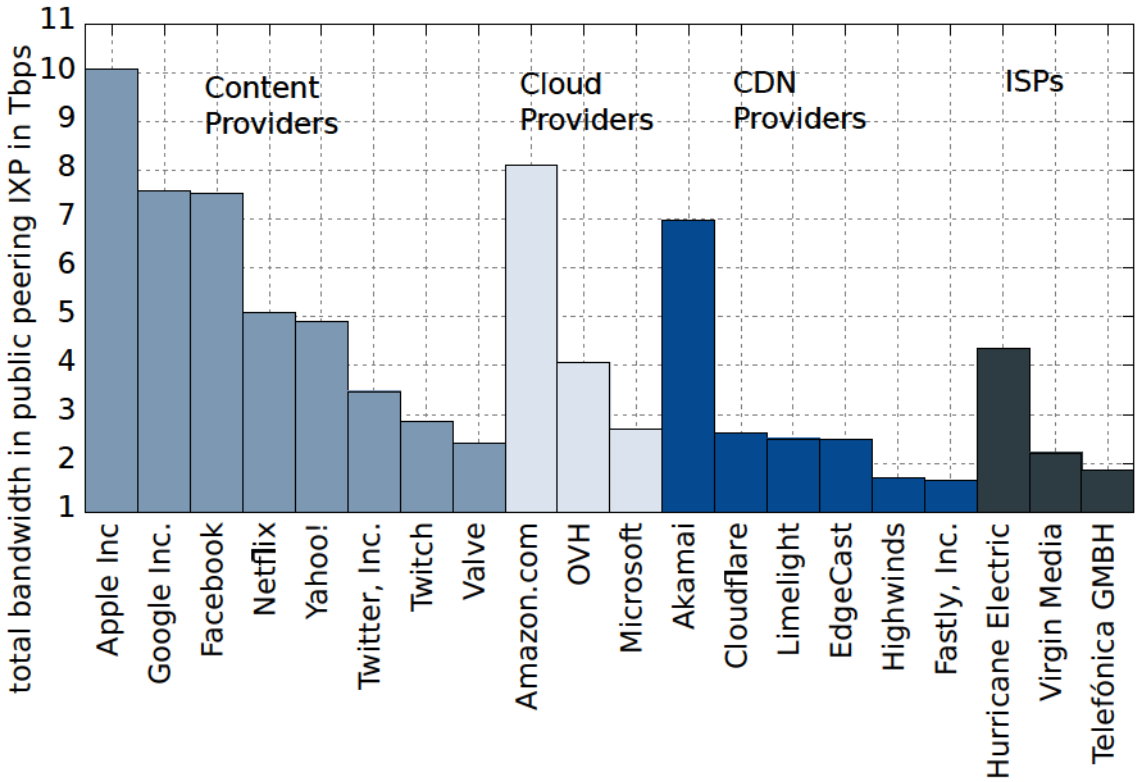


Figure 3.1: the 20 biggest Autonomous Systems per public peering bandwidth capabilities (Data exported from the PeeringDB API).

The amount of contents delivered through Internet is increasing at an exponential rate [Cisco, 2016], especially with the rise of online entertainment services for video and music such as Netflix or Spotify. The architecture of Internet, which always successfully adapted to new paradigms (such as cloud computing in the 2010s) is struggling to cope with this increase, especially since the End-Users (EUs) are asking for high quality of experience and ubiquitous access to content.

Traditionally, CPs rely on third party CDNs to distribute their service. CDNs build massive networks of cache servers deployed near the EUs to provide good performances, scalability and resiliency [Pathan et al., 2014]. Big Cloud Providers such as Amazon (CloudFront) or Rackspace (RackSpaceCDN) have also started implementing their own CDNs

services to expand their portfolio and provide their existing customers with means to deliver content. Large CPs (Netflix, Google) also started building their own delivery networks to reduce costs. By cutting out the middleman CDN providers, they also gain much more control over their delivery architecture and can roll-out specific enhancements through their native applications. For example, Spotify has been known to initially use a hybrid CDN/P2P solution [Kreitz and Niemela, 2010], with P2P accounting for 78% of the amount of remote data transferred to client. ISPs also started implementing so-called "Telco CDN " [Tuncer et al., 2013] with a view to regain more control toward the data going in their network through peering points and saving on transit costs.

Given the strong motivation for all parties (CDN, ISP, CP) to implement their own competing content delivery solutions, one can rightly ask what network softwarization can bring to the picture. In fact, the mutation has already happened, as we can see from Figure 3.1 where CPs networks have already exceeded ISP, CP and Cloud Providers in term of bandwidth allocated in peering points. We believe that virtualization of the Content Delivery service (especially in the ISP network) can provide several advantages that cannot be achieved by other means since they have not satisfactorily addressed three critical aspects at the same time: (a) how to leverage existing networks to reduce the required upfront investments, (b) How to implement custom delivery mechanisms tightly coupled with business proposal of CPs and (c) how to assure a global presence to target a worldwide audience.

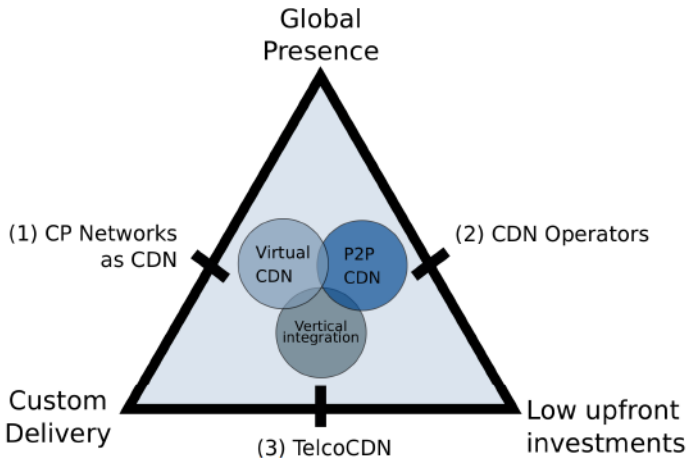


Figure 3.2: Trade offs for Content Delivery Networks Technologies.

In order to cope with those challenges, several solutions have been deployed in the industry, summarized in Figure 3.2.

1. **CPs build their own networks**: this allows the CP to customize its distribution mechanisms to suite its own business use-cases. Native applications (on mobile devices)

or custom browser extensions can be used to improve EU experience. The CP dimensions its network to suite its target audience (for example, Netflix expended its network as it opened its service to more and more countries [Adhikari et al., 2012]). This comes at a very high cost of owning the networks and managing their own fleet of cache servers.

2. **CPs purchase CDN Services**: CPs can rely on external CDN providers to assure the distribution of their contents. CDN Providers have extensively deployed their networks since the 2000s, and can provide connectivity to a global audience by themselves or by offloading part of the traffic through resources mutualization thanks to CDN Interconnection [Bertrand et al., 2012]. Using this model however prevents the CP to perform very specific changes in the way the content is distributed, as it relies on the licensed CDN technology that is out of its control.

3. **ISPs build TelcoCDN**: this model allows the tight integration between the ISP infrastructure and the content delivery technologies [Cho et al., 2011]. It may be possible to deploy specific algorithms, for example relying on Home Gateways assisted content delivery [Chellouche et al., 2012]. Additionally, building a TelcoCDN relies on already existing network infrastructures and Points of Presence, which makes the solution very cost effective. The drawback of this model is the locality of the content delivery: enhancements brought by TelcoCDN only apply to the ISP customers and it is hardly possible to build a global presence from a TelcoCDN.

As the above-mentioned challenges are poorly addressed by each of the existing solutions, several other alternatives have emerged driven by research:

- **Vertical Integration**: consolidating the role of a CDN, CP and ISP within a single entity can solve all the previously-mentioned problems. Having a technical control from the cache servers up to the EUs allows fine-tuning content delivery while reusing ISP infrastructure. Selling the content directly to the EUs through carrier-billing also keeps EUs captive of a unique provider for all their on-line entertainments. There are numerous examples of such complete or ongoing operations: Google with Google Fiber + Webpass, attempted Comcat and Time Warner merger, internet.org by Facebook. Even if those solutions are solving the technical problems, they raise antitrust issues [David McLaughlin, 2015].

- **P2P CDN**: this model is taking advantage of the well-known scalability advantages of peer-to-peer systems applied to online entertainment. Studies have shown that the model is viable [Luo et al., 2009] for live streaming and several companies[1] license their Peer-to-peer (P2P) technologies to complement legacy CDN deliveries [Yin et al., 2009], [Zhang et al., 2015]. The main advantage is that this model does

[1] e.g. peer5, streamroot

not require large servers and network deployments. Even if this technology is very interesting for live events, it is not clear, however, if it can be applied to Video-on-Demand use-cases, especially when users consume media that are not very popular. Given the fact that more and more users consume content on mobile devices, P2P CDN can have a negative impact on battery life. Even if "Machine Type Communication" use-cases are currently studied for the upcoming 5G standard, the main perceived use-case being Internet of Things (IOT)-type applications such as metering or payments [Shariatmadari et al., 2015], thus not necessarily targeting content distribution.

We believe that the emerging paradigm of network softwarization provides a new opportunity to overcome the above-mentioned challenges revolving around content distribution:

1. ISPs can leverage network softwarization in three ways. First by deploying NFV-compatible Points of Presences (such as NG-Point-of-presences (POPs) proposed in [Bertin et al., 2017]) capable of hosting third-party VNFs and sharing their customer hosting fees (like cloud providers do), hence creating a new revenue source. Second, CDNs implemented as VNFs also have the advantage of offloading ISP core network from content traffic, by serving them from the edge of the network. Third, this can be done by reusing ISPs' existing publicly subsidized optical fiber networks without the need to deploy yet another network.

2. CPs and CDN operators can develop and integrate their Content Delivery technologies as VNFs [Bouten et al., 2015] to be deployed in the ISP POPs, with a view to (1) fully customize their EU experience by implementing innovative content distribution technologies and (2) leveraging their proximity to EUs to propose first-in-class quality of service. Investing in the development of such VNFs is secured as NFV promotes reusing VNF transparently in any NFV-enabled infrastructure. Indeed, by specifying standard interfaces and semantics, this model addresses the classical vendor lock-in problems faced in Cloud Computing [Silva et al., 2013] and promotes code re-usability.

3. The market can benefit from low barriers to entry, since the development of VNF is purely based on software. New firms can operate traditional CDN services without having to invest in costly new networks.

After presenting the background on existing CDN solutions, the following sections of the chapter highlight the characteristics of our proposal: CDNaaS (CDN-as-a-Service), a Virtual Network Function for delivering content. We believe our work presents a first step toward a viable and pragmatic approach to deliver content from within the ISP network, in a way that is profitable to all actors of the content delivery ecosystem as a whole, including CPs and ISPs.

3.2 Background on existing CDN solutions

This section details what are the typical CDN architectures deployed and proposed by research and industry.

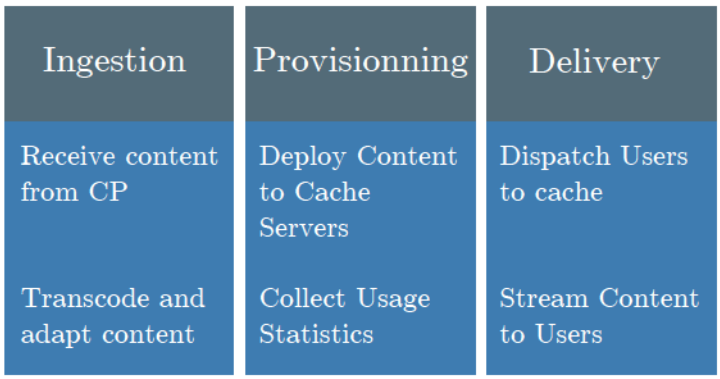


Figure 3.3: High-level functional breakdown of a Content Delivery Network Service

3.2.1 Standardization bodies

3.2.1.1 IETF

Several IETF RFCs describe the role of CDNs in various contexts. Early RFC 3040 [IETF, 2001] presents the taxonomy of web replication and caching infrastructure to reduce the response time and bandwidth consumptions. More Recently, CDNI [van Brandenburg et al., 2015] proposes a model for interconnecting several CDNs to expend their footprint (Section 4.1)

3.2.1.2 ETSI TISPAN

In [ETSI TISPAN, 2011], ETSI TISPAN specifies a standard CDN functional architecture for unicast content download and on-demand content. It outlines the three major stages of CDN as content acquisition, ingestion and deployment. This standard clearly defines the features for each module:

- The content deployment component is in charge of generating copies of content inside the CDN and coordinates the delivery and storage resources of replica.
- The request routing component is responsible for directing client requests for content to appropriate replica.
- The Content Delivery component is responsible for delivering content to the End-User.

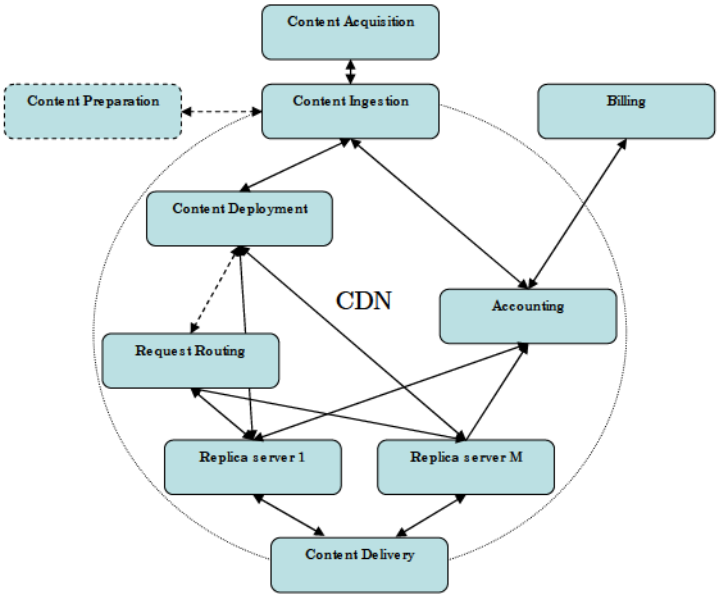


Figure 3.4: ETSI TISPAN functional architecture

- Billing uses inputs from the accounting module which monitors the CDN usage.
- Content acquisition handles how the CDN provider gets the content in the first place from the content source.
- Content ingestion tailors the content for delivery before leaving it up to the Content Deployment for provisioning.

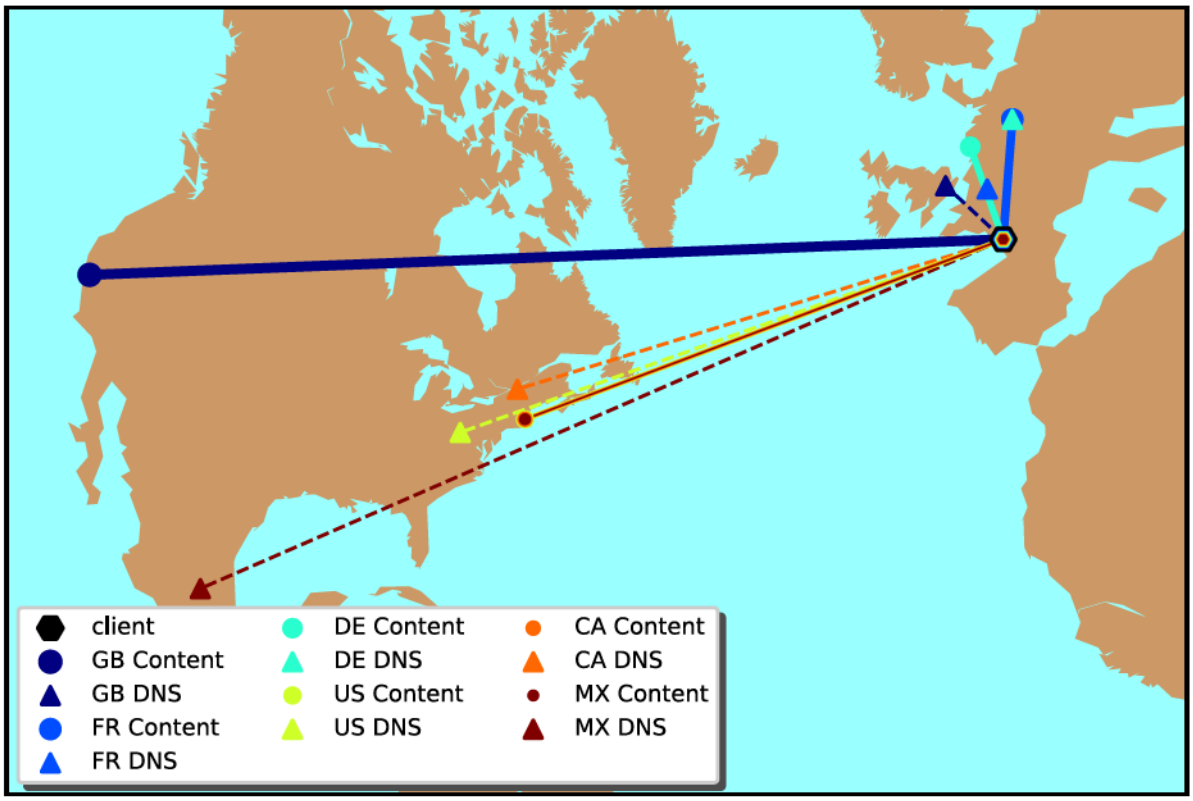
The CDNaaS proposal has taken the functional architecture from TISPAN as the basis for its requirements, as defined in Section 3.2.3.

3.2.2 Industrial implementations

In this section, we present the two main models for CDN implementation: pure-player CDN providers and integrated Content Provider CDN.

3.2.2.1 Pure-player CDN providers

This category regroups the network providers that market edge connectivity to content providers. Some large actors target a worldwide audience, such as Akamai [Nygren et al., 2010] or Cloudfare while others focus on specific regions, such as CDNlion for Asia or donweb for Latin America.

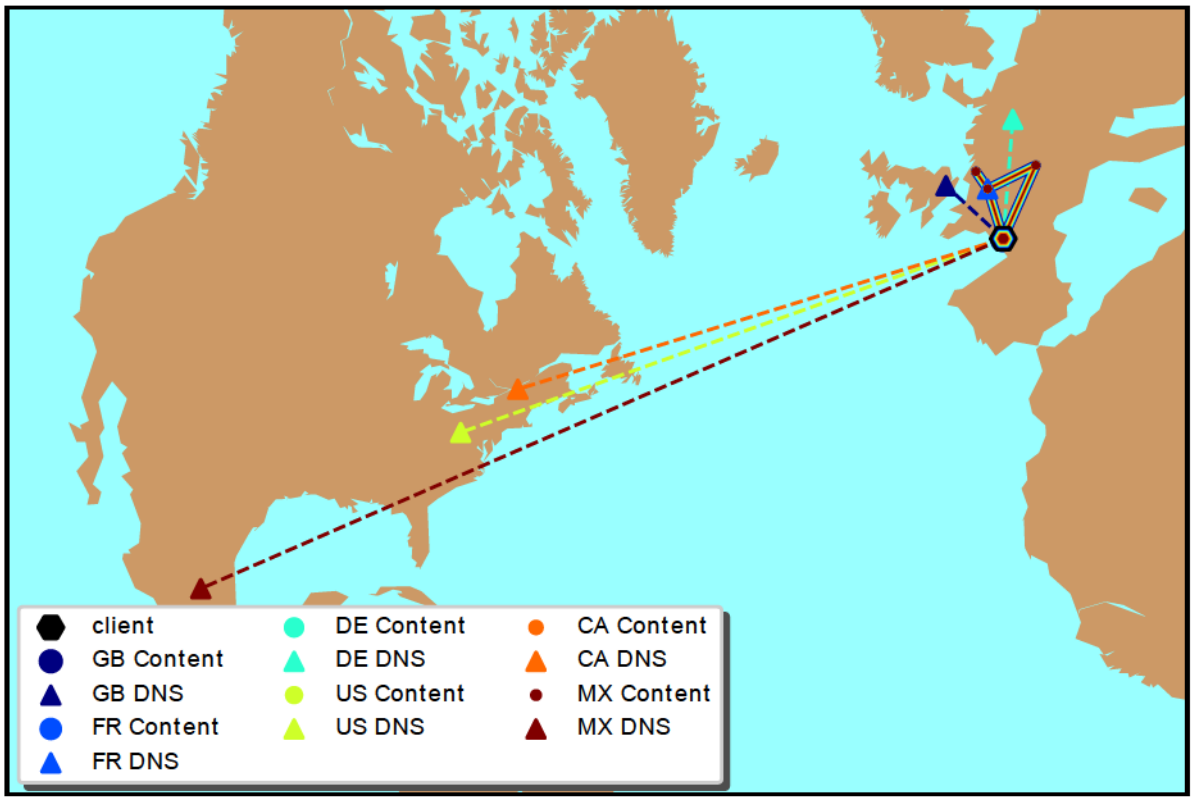


DNS Country	Content Country	End-User Country	DNS Content (km)	End-User Content (km)	AS Hosting the Content
CA	US	FR	388	5500	AS577 Bell Canada
DE	NL	FR	335	931	AS20940 Akamai International B.V.
FR	DE	FR	578	1039	AS16625 Akamai Technologies, Inc.
GB	US	FR	8616	9137	AS54113 Fastly
MX	US	FR	3664	5500	AS20940 Akamai International B.V.
US	US	FR	801	5500	AS16625 Akamai Technologies, Inc.

Figure 3.5: Traces of the CDN usage for popular home rental Airbnb, powered by DNS-based redirection CDN. We can see that the user being located in London is sometimes associated with servers located in the US when the DNS server is located in North America. Airbnb uses both Akamai and Fastly.

The CDN market is commoditized with rather standardized features; the competitors fight on price, global presence and on the number of POPs, which may be more a marketing argument than a scientific proof of superiority.

Considering the largest one, Akamai, its network is a virtual network overlay on top of the Internet, which supports the distribution of various types of content (including also dynamic contents). It is composed of 5 main modules. The **Mapping System** maps an URL to a **edge server** present in one of the **edge platforms**. Content is retrieved from origin



DNS Country	Content Country	End-User Country	DNS Content (km)	End-User Content (km)	ASHosting the Content
CA	FR	FR	5455	12	AS15557 SFR (from Google Cache 3s-sfr-bod3)
DE	FR	FR	1043	12	AS15557 SFR (from Google Cache 3s-sfr-bod3)
FR	FR	FR	504	12	AS15557 SFR (from Google Cache 3s-sfr-bod3)
GB	FR	FR	742	12	AS15557 SFR (from Google Cache 3s-sfr-bod3)
MX	FR	FR	9134	12	AS15557 SFR (from Google Cache 3s-sfr-bod3)
US	FR	FR	6282	12	AS15557 SFR (from Google Cache 3s-sfr-bod3)

Figure 3.6: Traces of Youtube server selection for the most popular video in the world (Despacito with more than 3 billion views, cached in every Google Cache Server). The server selection is independent of the DNS server used and only relies on customer information.

through a **transport system**, which is the Akamai backbone network allowing edge caches to retrieve content, not from origin but from parent caches, making sure that the CDN is not overloading the origin server. The platform is also equipped with various **data collectors and analytics system** to supervise the operation on the network and a **Management System** allowing the customer to derive KPIs and audience information.

Akamai uses DNS-based request routing to integrate with their customers content. Customers serve their content through dedicated hostnames, which are resolved by the Akamai load balancer to an Akamai Server. The server is selected based on proprietary

criteria, such as the availability of content and the server load. In order to improve end-user quality of experience, the closest server is selected to minimize the round-trip time. This mechanism takes into account the location of the DNS server hit by the end-user to infer its localization on the Internet [Pathan, 2014b]. This method has a major drawback: in their paper [Krishnan et al., 2009a] Google explains that traditional latency-based redirection can be rendered ineffective in a couple of scenarios such as using distant DNS nameserver and that decisions made at the granularity of prefixes can be sub-optimal. To illustrate this issue, Figure 3.5 shows an experiment[2] where contents are downloaded from the popular global website Airbnb. This service relies on a multi-CDN solution, using among others Akamai and Fastly. A direct correlation appears between the DNS server used for the query and the location of the content server. For example, even if the client is located in France, the sever selection algorithm establishes that the optimal server is located in the US whenever the chosen DNS server is in North America. For this reason, today major content providers deploy their own CDN and compute the user-server assignment directly on their servers based on the IP of the end-user.

3.2.2.2 Content Providers owned CDNs

Large Content Providers, such as Netflix or Google with Youtube, which can afford building their own distribution network gather important advantages over others relying only on pure player CDNs. First of all, by cutting the middleman, they are able to save on costs, supporting the delivery network with their own appliances or cloud infrastructure. They are also capable of optimizing their delivery service to improve their customers' quality of experience by developing specialized software. For example, Netflix has invested in Open-source software and owns 130 repositories (such as Hystrix for resilience in complex distributed systems, eureka for Amazon Web Service based resiliency and failover, and SimianArmy for Cloud resiliency) totalizing more that 13K forks[3]. Google developed royalty-free video format for browsers such as WebM, VP8 and VP9[4] and supports its video service through their Google Global Cache CDN.

Leveraging on their software assets and integrated infrastructures, large cloud providers are able to support more advanced user-server assignation methods, based on the IP address of the client, its user-agent and the URL of the content which is processed from the HTTP request. Figure 3.6 shows how the DNS server location is not used when retrieving a video from Youtube; the URL of the content from a Google Global Cache is returned instead by the HTTP Server.

[2] A dataset containing the same results for the 15 top websites according to Alexa Internet Inc from 20 locations around the world is available online http://data.nextnet.top/cdn/

[3] https://gist.github.com/nherbaut/878448e781eed85b011db6112306481a

[4] https://www.webmproject.org/

3.2.2.3 Virtual CDN concept

Research papers such as [Frangoudis et al., 2016], [Frangoudis et al., 2017], [Yala et al., 2016] and the SONATA European Project[5] have addressed the Virtual CDN concept. By prototyping the configuration and provisioning of data within a TelcoCloud environment, these works shade some light on the feasibility of running a CDN in the cloud.

Little work has been carried out to describe the particular integration challenges with an NFV platform. The deployment of a CDN as a VNF has been initially described in ETSI Use-cases [ETSI, 2013] and has been demonstrated [Kim and Lee, 2014] within the ETSI NFV Proofs of Concept Framework[6]. The concept was also used in [Giotis et al., 2015] to support a proposed schema of Policy-based Orchestration of VNFs. We complement this previous work by directly addressing and evaluating the deployment challenges on a real NFV platform implementation. To guide us through the evaluation, we start by expressing the requirements for CDNaaS in the following section.

3.2.3 Requirements

To clarify our design goals, we established a list of requirements derived from [ETSI, 2013a] and the previous cited work, presented in Table 3.1. Each requirement is evaluated based on architectural considerations or with the validation of the feature on the T-NOVA testbed. The next section presents the CDNaaS approach and technical architecture.

3.3 The CDNaaS proposal

As we saw from previous sections, the idea of creating a virtual CDN is emerging for four main reasons. Indeed, CDN concept benefits from dynamic capacities to accommodate fluctuating traffic. Thanks to the scalability offered by cloud resources, virtual CDN can adapt to the demand in term of number of available system (Storage, CPU and RAM) and network resources. Moreover, CDN placement is also crucial to provide a good Quality of Experience (QoE) to end-users. On-demand edge cloud resources allow CDN provisioning to adjust to local demand, even for short-lived period. Furthermore, CDN features can be enhanced rapidly by rolling out novel algorithms and custom implementations, faster than with hardware appliances. Finally, virtual CDN is supported by cloud providers' infrastructures that can be lease on demand, without large upfront costs.

CDNaaS aims at offering a virtual CDN as a service while benefiting from all the advantages of NFV concept. First, by being NFV Management and Orchestration (NFV MANO) compliant, CDNaaS has the assurance that the function is reusable on any ISP platform supporting NFV. Second, thanks to T-NOVA Marketplace, vCDN customers can express their

[5] http://www.sonata-nfv.eu
[6] https://goo.gl/jKGxcz

Table 3.1: Requirements for CDNaaS

	Requirement	Justification
	Management	
1	CDNaaS shall be integrated within a NFV architecture (Section 3.5.1)	the goal is to build a Service deployed in a Cloud, which can be managed through defined NFV interfaces
2	The vCDN customer shall be able to manage CDNaaS through high level interfaces (Section 3.5.2)	CDNaaS should not only be benefiting Telcos, but also managed as a service by a CDN operator or a CP. For this, the vCDN customer cannot have full visibility on the platform and network since these data can be Confidential.
	Functional	
3	Request Routing decisions shall use information from HTTP transactions (Section 3.3.2)	DNS-based redirection have limits which encourage shifting to HTTP-based systems
4	Content Storage shall support location-aware storage nodes (Section 3.3.3)	Edge caching is particularly efficient at reducing latency when the cache server is located near the en end-user
5	Content Ingestion shall be able to accept common media formats and produce standard and enhanced new formats (Section 3.3.4).	There's a particular challenge in managing CPU-intensive operations such as content trans-coding
	Non-Functionals	
6	Internal CDNaaS onboarding, deployement, configuration, and scaling shall be automated (Section 3.4.1)	Automation speeds up the development process and quality of the software
7	All the components of CDNaaS shall scale automatically (Section 3.4.3)	To adapt to any kind of workload, CDNaaS must be able to increase its capacities

connectivity needs through high level Service Level Agreements (SLAs). Third, CDNaaS open design allows third-party VNFs to be used to improve performances.

Following the functional architecture shown in Figure 3.4, CDNaaS architecture can be broken down into 5 modules, as detailed in Figure 3.7:

- The **VNF Controller** (VNFC) is responsible for the external communication of CDNaaS and manages both the *T-Ve-Vnfm* interface toward the VNF Manager and the *T-Nf-Vi*

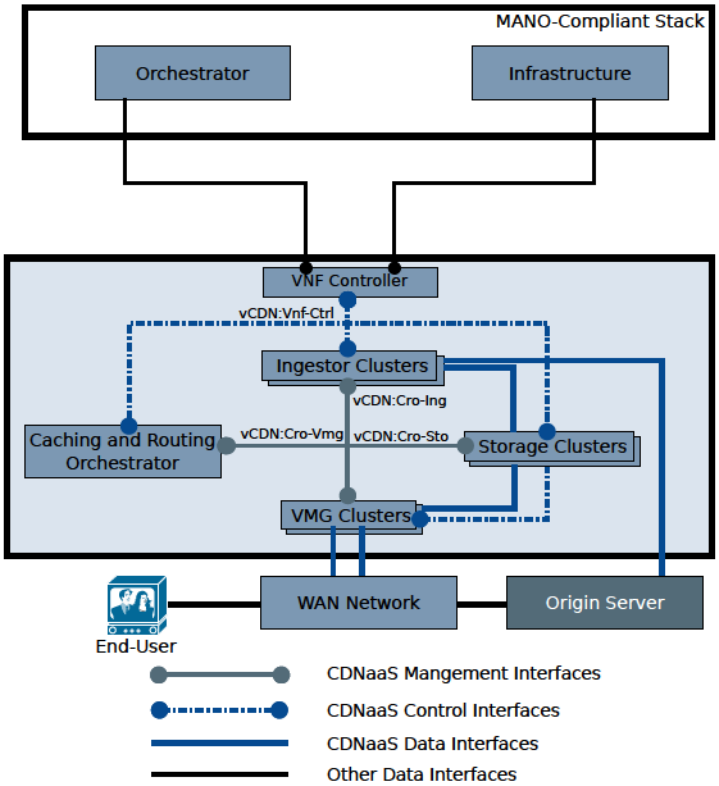


Figure 3.7: High Level Architecture of CDNaaS

one toward the VIM used for monitoring. It also centralizes the configuration of the other modules and plays the role of internal technical orchestrator of the VNF through the *vCDN:Vnf-Ctrl interface*.

- The **Caching and Routing Orchestrator** (CRO) is the module responsible for enforcing the caching configuration (what to cache and where) and the requests routing configuration (which users should be directed to the VNF Storage). It configures the other modules through the dedicated *vCDN:Cro-Vmg*, *vCDN:Cro-Sto* and *vCDN:Cro-Ing* interfaces.

- The **Ingestor Clusters** are in charge of downloading the content from the original server, which stores the content requested by the end-user, processes it according to the configuration, and sends it to the Storage Clusters nodes.

- The **Storage & Streaming Clusters** are responsible for storing the content and making it accessible to CDN end-users, through HTTP protocol.

ETSI TISPAN Architecture	**CDNaaS + T-NOVA Architecture**
Content Delivery and Replica Server	Storage Clusters @ NFVI-POP
Request Routing	VMG Clusters @ NFVI-POP
Accounting / Billing	T-NOVA Marketplace through *T-Nf-Vi, T-Vi-Vnfm, T-Ac-Or* interfaces
Content Ingestion/Content Preparation	Ingestor Clusters @ NFVI-POP
Content Acquisition	T-NOVA WAN Configuration through WICM

Table 3.2: ETSI TISPAN and CDNaa+T-NOVA Functional Architecture Comparison

- The Virtual Media Gateway Clusters (**VMG Clusters**), which role is to perform the user-server assignment by analyzing HTTP requests from the end-users. They also can analyze the traffic from the original server and publish content usage to the CRO.

3.3.1 Comparison with TISPAN architecture

Figure 3.7 shows that the CDNaaS functional architecture is compatible with the ETSI TISPAN (Figure 3.4) on several levels. Table 3.2 shows that most of the TISPAN CDN functional requirements are covered by internal CDNaaS VNFC. Some non-fonctional requirements such as accounting and billing are brought by T-NOVA Dashboard and the Network configuration is brought by the T-NOVA WICM module.

Let us now dig deeper into the implementation of each CDNaaS VNFC to understand how functional requirements 3,4 and 5 are handled.

3.3.2 The Virtual Media Gateway

The Virtual Media Gateway is a configurable HTTP Proxy that handles both requests and responses from end-users. It can be configured through a REST Application Programming Interface (API)[7] with features specific to the vCDN business. The design philosophy of the Virtual Media Gateway has been inspired by similar-to-SDN forwarding devices. Indeed, the VM can be configured with a `filter` that triggers an `action` in case of a match. The difference, however, is the applicable domains of the VMG, in a sense that filters can be applied only at HTTP level and actions taken are HTTP calls or responses.

3.3.2.0.1 Filters and actions

Filters can be applied to both HTTP requests and responses. They can support every standard HTTP features, such as Protocol Level, URL and Headers. Actions are triggered

[7] http://data.nextnet.top/vmg-api-doc/

Table 3.3: REST API used to configure the VMG Filters and Actions

Description	HTTP Verb	REST Payload

Notify the CRO when a response is of type webm	POST	`filter:`
		`response:`
		`headers:`
		`Content-type: video/webm`
		`action: notify`

Forward to the Storage5 for a content URL	POST	`filter:`
		`request:`
		`url: http://video.com/bbb.mp4`
		`action:`
		`forward : pop5`

Forward to the Storage1 requests for identified users	POST	`filter:`
		`request:`
		`and:`
		`url: http://video.com/*.mp4`
		`user:`
		`identified: true`
		`action:`
		`forward : pop5X`

Remove the action corresponding to the filter where users are identified	DELETE	`filter:`
		`request:`
		`user:`
		`identified: true`
		`action:`
		`forward : *`

if filters are verified. Actions can be of two types: `forward`, which forwards end-user's requests to the provided NFVI-POP and `notify`, which notifies the CRO of the request performed by an end-user. Filters and actions can be configured through a REST API, as illustrated in Table 3.3.

3.3.2.0.2 Implementation

The Virtual Media Gateway is implemented in JAVA and uses the Netty Framework. Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers and clients. One of the most striking features of Netty is that it can access resources in a non-blocking approach, meaning that some data is available as soon as it gets in the program. This avoids wasting system resources while waiting for the content to become available; instead, a callback is triggered whenever data is available. This also saves system resources by having only one thread for resource monitoring. Netty is one of the building blocks used to implement the vMG network capabilities. Netty is used by well-known SDN Controllers such as Floodlight, ONOS and ODL[Shalimov et al., 2013].

3.3.2.0.3 Scalability and Availability

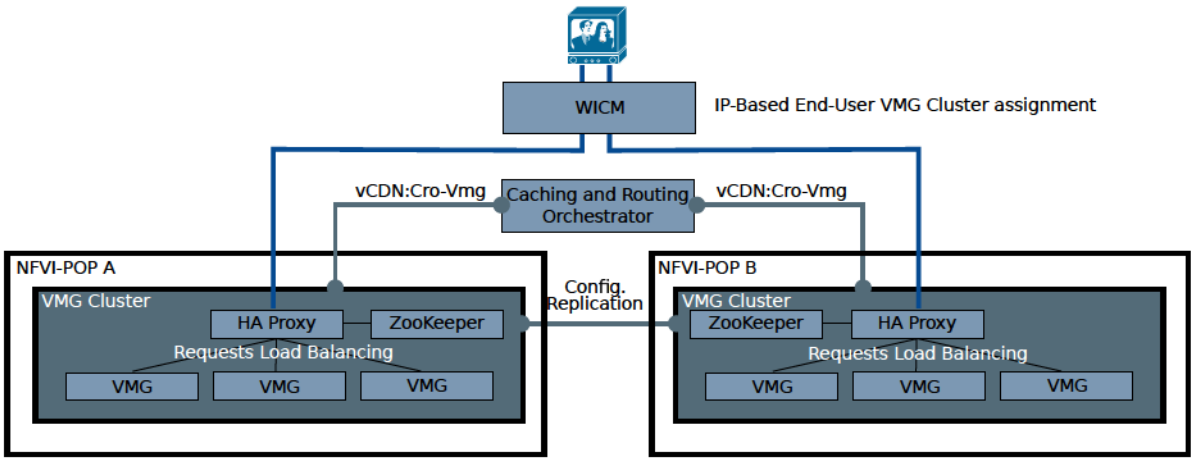


Figure 3.8: Internal Architecture of VMG clusters

Figure 3.8 shows that VMGs are grouped in clusters, each cluster belonging to a different NFVI-POPs. When the end-user sends a request, the T-NOVA WICM is responsible for assigning it to the NFVI-POP. This can be done by assigning each POP a different IP prefix, depending on the outcome of the Service Mapping Algorithm. Once the request reaches the appropriate NFVI-POP, it is received by a Loadbalancer (HAProxy to load-balance queries [Kaushal and Bala, 2011]), that splits the requests in a round-robin fashion to the VMGs.

The Caching and Routing Orchestrator deploys the same configuration to each cluster. This configuration is deployed in a Distributed Key-Value store (ZooKeeper [Hunt et al., 2010]). Zookeeper is in charge of providing the configuration to each VMG to make sure that the instances are synchronized.

Requirement 3 ✓ This section outlined a programmable network component able to support HTTP payloads inspection and to apply redirection and notifications rules.

By using well-known tools from the Cloud Computing world, we are able to bring scalability and performance to the Virtual Media Gateway for the configuration deployment aspects. However, these properties do impact the real throughput of the network function, as evaluated in Section 3.5.3. In the following, the architecture for the HTTP-capable distributed storage solution, i.e., the vStreamer, is presented.

3.3.3 vStreamer

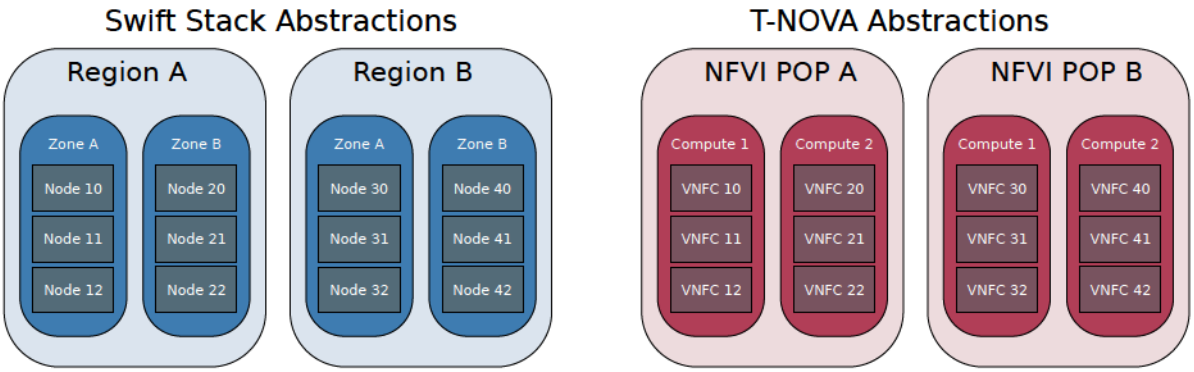


Figure 3.9: Comparison between T-NOVA and SwiftStack abstractions

CDNs can be used to store Terabytes of data that can be spread among several data-centres and that must be tolerant to failures. They also need to scale and must support adding or removing storage nodes, as defined by the scaling policy. To this end, we decided to deploy an instance of Swiftstack[8], an opensource project, which proposes to create a cluster of storage nodes to support Scalable Object storage with High availability, Partition Tolerance, Eventual Consistency and HTTP Access (other alternatives like Ceph were also considered). Swift storage is composed of a *distributed hash table* ("Ring") storing the location of contents and *objects* which actually store the data content. While objects are stored on *Storage Nodes*, a *Swift Proxy* is used to maintain the state of the nodes and enforce the storage policies. Swift has several logical abstractions that spread the objects into Nodes, the Nodes into Zones and the Zones in Regions, as shown in Figure 3.9. These abstractions can be transposed in T-NOVA, where each Region can be seen as a different NFVI-POP, each Zone is a Compute Node and each Node is a VNFC hosting a Storage Node.

To support our vCDN storage use-case, each object present needs to be stored in every POP, in order to minimize the latency toward end-users. Additionally, for resiliency purposes, the VNFCs need to be hosted in a different compute node, such that if a compute node fails, then not all the VNFCs are impacted and the service can still operate. To do so, swiftstack configurations were implemented with the following rules:

[8]https://www.swiftstack.com/

- Each object must be stored at least in each Region.
- For each object present in a Region, it must be stored in at least 2 different Zones.
- For each NFVI-POP, at least half the Storage VNFC must be located in 2 different compute nodes[9].

Requirement 4 ✓ Thanks to the adaptation of the NFV concepts to the abstractions provided by swiftstack, we were able to build a network component that can store data in a distributed fashion, providing CDNaaS with a powerful distributed storage engine.

3.3.4 Content Ingestion

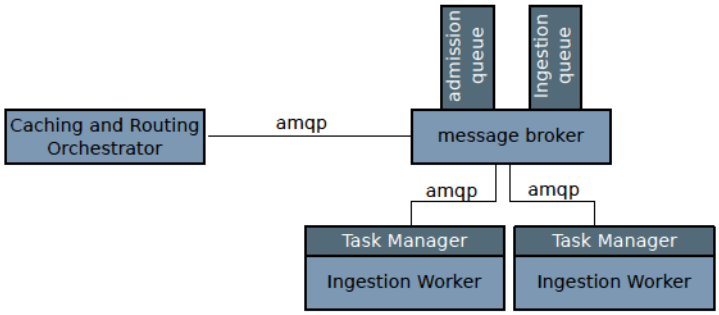


Figure 3.10: Architecture of the Ingestion

The ingestion of content within the system allows storing content in a format better suited for on-line distribution. Such formats or *profiles* use specific codecs, framerate, encapsulation and segmentation (for adaptive streaming). During the ingestion phase, the following operations are performed:

1. Download from the original source server;
2. Temporary storage for processing;
3. Analysis of the current format, resolution, framerate and bitrate;
4. For each supported output profile:
   a) Transcode the video using ffmpeg;
   b) Prepare the video by creating DASH and HLS segments;
   c) Publish the video in the appropriate Storage Node;

[9]This rule was described in the specifications, but never implemented in T-NOVA

5. Notify the Caching and Routing Orchestrator that the ingestion cycle is complete.

Several tasks are synchronous (1,2,3,5) while others can be parallelized (4.a.b.c.). For this reason, we chose to use an asynchronous job model as shown in Figure 3.10, where worker nodes are connected to a scalable message broker through AMQP[10] and handle ingestion orders sent by the CRO. Two queues are used in the message broker. First, the admission queue handles the download of the original content in a temporary workspace, before dispatching ingestion orders in the ingestion queue. Second, the ingestion queue is monitored by the ingestion workers that carry out the computer intensive tasks of video transcoding and segmentation.

Requirement 5 ✓ In this section, we described a module that uses a distributed jobs framework to implement a complex transcoding and adaptation workflow.

The next section is dedicated to the study of non-functional requirements, where we make a case for using Linux Containers to support an improved development model, easier orchestration and scalability.

3.4 Linux containers benefits for CDNaaS deployment

In this section, we explain how we leveraged Docker Linux Containers for the internal orchestration of CDNaaS. The stateful nature of this VNF and the necessity to support scaling-out by a 10 or 100 factor within minutes compelled us to use a container-based clusterable microservice architecture where a simpler monolithic approach would not have been possible. We explore software engineering problems of microservice deployment, scaling and container-based resiliency.

3.4.1 An agile VNF development cycle

As seen in the previous section, three out of four functional modules of our VNF are based on clusters of resources. Managing 3 clusters based on 3 different VM base images can be very time consuming from a development perspective. Indeed, the typical development cycle observed from T-NOVA industrial partners included the tasks shown in Figure 3.11.

This process forms a V-cycle, in the sense that before being able to fix or improve a feature in the VNF, one needs to replay all the intermediary steps that are prone to errors and do not bring any added value.

For this reason, we created an agile model relying on Docker Containers to automatize the process. Instead of deploying our code in Virtual Machines, we use a single VM image containing the base OS and a configuration management tool deployed in the VNF Controller. The configuration of the VNF and the service is done once for all, and code is injected during a bootstrapping phase right before the VNF is started. When the VNF is bootstrapped,

[10] the ingestion was implemented with RabbitMQ

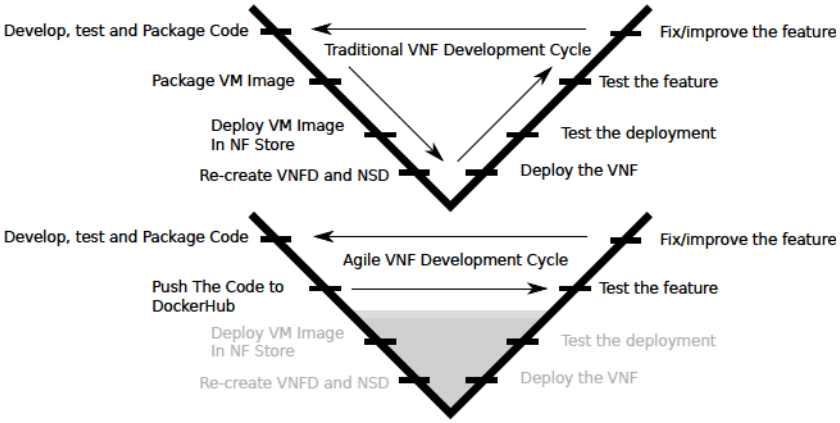


Figure 3.11: Comparison between the Traditional VNF Lifecycle and the agile lifecycle

the VNF Controller retrieves the infrastructure descriptor from our github repository. This descriptor[11] is used by the configuration management daemon[12] to determine what components need to be installed on each VNFC. The components are subsequently downloaded from a container repository (DockerHub[13]) and run as Docker Containers. Thanks to this model, we were able to deploy and test new software without any intervention on the NFV infrastructure side. Software quality was assured thanks to container-based continuous integration. Thanks to the minuscule runtime overhead of Linux containers [Felter et al., 2015], performances for our custom modules is close to native.

Requirement 6 ✓ Using Containers and a configuration management tool, we fully automatized the development life-cycle, from code production to VNF deployment.

3.4.2 Dual Orchestration using Linux Containers

The interaction between the orchestrator (through the VNF Manager) and our VNFs occur through standard life-cycle events, such as stop, start, scale-out, scale-in and configure. The format of these events are the same for each VNF and are too general to precisely pilot the internal architecture of each VDU. For this reason, we created a Dual-Orchestration Mechanism, by separating the concerns of the NFV orchestrator (acting at the VM level) and an internal orchestrator (acting at the Container Level, run by the VNF Controller). The Goal of the NFV Orchestrator consists in creating the VNFC from the base image and configuring its external network accordingly. The rest of the configuration is assured at the internal orchestration level which deals with the following tasks:

- Launch the right number of containers for each VDU.

[11] https://github.com/dngroup/t-nova/tree/master/salt
[12] https://saltstack.com
[13] https://hub.docker.com/

- In case of container failure, relaunch it with an exponential backoff.
- If a new VNFC is provisioned due to a scaling-out event launched by the TeNOR Orchestrator, deploy the supplementary containers on it and configure the underlying clusters.

By decoupling the orchestration of the service and the VNF from the internal orchestration of the VNF internal components, we assured an easier integration to other NFV platform (since we are only relying on high-level lifecycle events). In our implementation, such dual-orchestration mechanism is deployed within the VNF itself. Another possibility would have been to develop a custom VNF Manager (based on our existing VNF Controller) to have more fine-grained, application aware life-cycle events, as proposed in the SONATA EU Project [SONATA Consortium, 2015]. While this solution is tempting and more in line with the ETSI Mano architecture, it adds the complexity of deploying custom code inside the NFV architecture, more difficult task than installing it within the VNF itself. More in-depth analysis of this system can be found in T-NOVA deliverables [T-NOVA Consortium, 2016b] and [T-NOVA Consortium, 2017].

3.4.3 Container-based scaling

In this section, the concern is placed on scaling, following the anti-fragile pattern [Abid et al., 2014], allowing each micro service to operate in autonomy so that the whole VNF can be resilient and easily scalable.

First, the monitoring agent reports that the storage is running low, or that the ingestion procedure cannot cope with the ingestion demand, thanks to application-specific metrics. The NFV Orchestrator then starts a new VM (storage node, ingestion node or VMG node depending on which metric is breached) and this node is provided with the IP address of the VNF controller. The new VM connects to the VNF controller so that it can be part of the infrastructure under its supervision. When the VNFM sends the scale-up lifecycle event, the controller re-synchronizes the infrastructure with all the connected VMs.

The same procedure is used when scaling the VM and when starting them, making the process repeatable: the VNF Controller downloads container images, configures the VM (configuration files are written and services are launched) according to its "role" (storage node, ingestion node or VMG node). Subsequently, depending on the role of the new VNFC, several configuration steps are taken by the VNF Controller:

- Once a new ingestion node joins the pool, it is provided with the IP address of the message broker and starts consuming messages;
- Once a new storage nodes starts, the storage proxy node re-computes the distributed hash tables and re-balances the load amongst the nodes automatically;

- For the VMG nodes, Haproxy is reconfigured with the Node IP and starts balancing the load to the new VMG instance.

Requirement 7 ✓ Thanks to the dual-orchestration mechanism and the scalable by design architecture of CDNaaS internal components, the scaling and deployment procedures were fully automated.

3.5 CDNaaS Integration and Validation

In this section, we describe the integration of CDNaaS in the T-NOVA project.

First, we explain the role of each standard NFV MANO interfaces used during the onboarding and VNF operations, as well as the specific T-NOVA interfaces used for service provisioning and SLA configuration. Finally, we present experimental results obtained on the T-NOVA distributed testbed for each components: content distribution and content ingestion.

3.5.1 Integration within T-NOVA

To fulfill the first requirement, we integrated CDNaaS in a MANO-compliant NFV stack, T-NOVA. Figure 3.12 represents a simplified version of the integration of CDNaaS into a NFV architecture. T-NOVA interfaces have been presented in Section 2.3 and are detailed in the project deliverable [T-NOVA Consortium, 2015]. The Marketplace, Orchestration and NFV Infrastructure Management components are vanilla software and are used by CDNaaS and by other VNFs developed by the partners. They are aligned with the ETSI Mano [ETSI, Network Functions VirtualisationV., 2014] Interfaces and extend them notably with the inclusion of the Marketplace component that pilot the composition of service level agreements and financial reporting. CDNaaS and Network Management System (NMS) are components developed specifically for the content delivery use case.

Marketplace-Orchestration Interfaces

- **T-Sl-Or**: The SLA management module sends the SLA agreement to the Orchestrator. The Orchestrator communicates SLA-related metrics to the management module so that the latter can check SLA compliance.

- **T-Ac-Or**: The Orchestrator sends the Accounting module with all the service track information.

Orchestration-VIM Interfaces

- **T-Vi-Vnfm** is responsible for the exchange of infrastructure monitoring information allocated to the VNF either through explicit request by the VNF Manager or through periodic reporting initiated by the VIM.
- **T-Or-Vi** allows the Orchestrator to request reservation/allocation of resources and NS-related lifecycles operations and for the VIM to report the characteristics, availability, and status of infrastructure resources.
- **T-Or-Vi** allows the Orchestrator to request reservation/allocation of resources and NS-related lifecycles operations and for the VIM to report the characteristics, availability, and status of infrastructure resources.
- **T-Da-Or** is used by the customer and the ISP to get monitoring information from the service.
- **T-Or-Tm** supports requests from the Orchestrator to provide connectivity services typically for inter-data center or connection to customer services (in our case, connection to the vCDN Customer Source Server).

VIM-CDNaaS Interfaces

- **T-Nf-Vi** dispatches management decisions from the VIM to the Compute domain and communicates back the Compute domain status and monitoring metrics.
- **T-Tm-Tr** configures access to the WAN, such as providing connectivity from the edge of the NFVI-POP down to the VNFs. In our testbed, this interface is in charge of deploying Openflow rules to our SDN-capable Forwarding device.

Orchestration-CDNaaS Interfaces

- **T-Ve-Vnfm** interface allows the VNF Manager to configure the VNF, collect monitoring/performance data, and be notified about faults within the VNF.

Element Manager Interfaces

- **Ve-Vnfm-em**: in [ETSI, Network Functions VirtualisationV., 2014] ETSI defines this interface used for exchanges between the Element Manager and VNF Manager, and supports instantiation, high level run-time information retrieval, configuration update. In T-NOVA, the *Vf-Vnfm-em* interface, however, is limited to configuration update, as the VNF lifecycle is bound to the network service Lifecycle which is controlled by the Marketplace, accessible to the vCDN Customer through the Dashboard.

- **vCDN:emAPI** : The CDNaaS element manager is a specialized software that allows the vCDN Customer to monitor and configure the behavior of CDNaaS. CDNaaS-specific configuration items are sent to the VNF Manager which "tunnels" the commands to CDNaaS through T-Ve-Vnfm. The vCDN:emAPI is not used to control low-level metrics such as CPU, Network or disk access. Instead, the monitoring interface *T-Nf-Vi* collects such data, and trickles them up to the Orchestrator which handles VNF-scaling to adapt the VNF capabilities to the adequate level according to the SLA.

Requirement 1 ✓ As we saw from the details of the interfaces and the modules of the system, CDNaaS is integrated with an architecture that is compatible and extends the one envisioned by ETSI. We now present more in details the initialization and configuration workflow of CDNaaS, and explore how high level interfaces are used to configure the service.

3.5.2 CDNaaS instantiation and configuration workflow

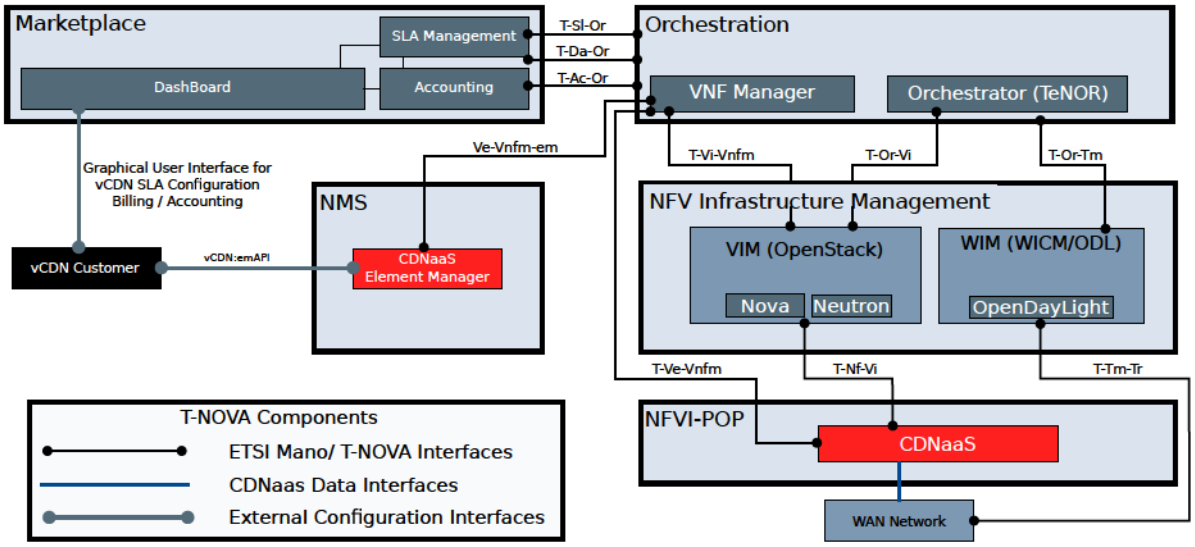


Figure 3.12: High Level architecture for a CDNaaS running in a NFV-as-a-Service platform

To better understand the interactions between the different modules composing CDNaaS, ETSI MANO and T-NOVA architecture [T-NOVA Consortium, 2017], Figure 3.13 shows a sequence diagram corresponding to the configuration, instantiation and management of a CDNaaS instance bought by a vCDN Customer.

The first step consists for the **(1)** vCDN customer to configure its vCDN service through the Markeplace Graphical User Interface (GUI). In this interface, the customer can select different SLA levels depending on its need (number of concurrent users, storage size and

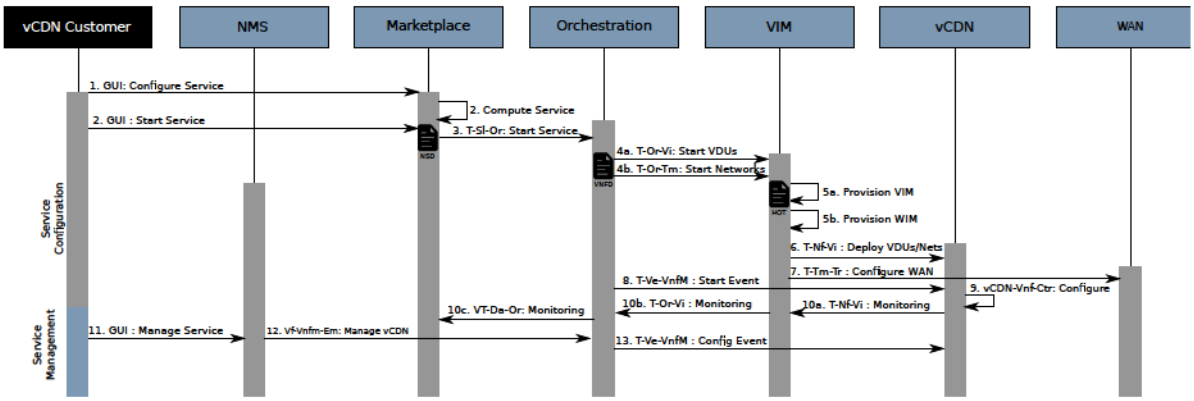


Figure 3.13: Sequence Diagram of CDNaaS, the vCDN Customer interacts with the T-NOVA Marketplace to create the service, and with the Element Manager for Service Configuration and monitoring.

ingestion capacities). For example, a "bronze" level SLA could correspond to 500 simultaneous users with a storage of 10TB. The SLA is translated to a Network Service Descriptor (NSD) to be processed by the Virtual Infrastructure Manager (VIM).

Next, the **(2)** vCDN Customer Starts the service through the Marketplace GUI, which causes the **(3) NSD** to be sent to the orchestration layers through the *T-Sl-Or* interface. The NSD represents a service, which is composed of one VNF or more chained together. Each configured VNF is described in a VNFD file. A VNFD is the template which describes a VNF in terms of deployment and operational behavior requirements. The VNFD also contains connectivity, interface and KPIs requirements that are used by NFV-MANO functional blocks to establish appropriate Virtual Links within the Network Function Virtualization Infrastructure (NFVI) between VNF Components instances, or between a VNF instance and the endpoint interface to other Network Functions.

When the orchestrator has generated the VNFDs, they are sent (steps **4a** and **4b**) to the VIM components through the *T-Or-Vi* to configure the Virtual Deployment Units (VDU) and their Networking. A VDU supports the description of the deployment and operational behavior of a subset of a VNF, or the entire VNF if it was not componentized in subsets [ETSI, 2013b]. In our case, as the VIM is built on top of Openstack with Hypervisor-backed compute nodes, so each VDU is implemented through a Virtual Network Function Component Instance (VNFC Instance) corresponding to a Virtual Machine (VM).

Steps **(6)** shows the creation of CDNaaS Virtual Deployment Units (VDUs) and associated networks through Interface *T-Nf-Vf*. Each VNF Descriptor is converted to the native Openstack orchestration module Heat[14] Orchestration Template (HOT file) which is used

[14]https://github.com/openstack/heat

to effectively create Nova[15] Server Instances, Neutron[16] router and networks, Storage resources and so on. The external connectivity **(7)** of the VNF and its potential placement into a service function chain is configured by the WAN Infrastructure and Connectivity Manager (WICM) thanks to Netfloc, as OpenDaylight plugin that carry out Service Function Chaining (SFC) based on virtual Mac addressing [Trajkovska et al., 2017]. The SFC configuration can take place between several Points of Presence, and all the VNFs of a chain do not need to be collocated.

Steps **(8-9)** are dedicated to the configuration of CDNaaS. Indeed, even if several VDUs are instantiated and connected in a Network Function Virtualization Infrastructure Point of Presence (NFVI-POP), they need to be given the network addresses of each other and be told to start accepting connections. Next, specific runtime parameters have been configured by the vCDN Customer through the Marketplace GUI, and need to be passed to CDNaaS for configuration. Once CDNaaS is started, it connects to the Monitoring module of the VIM to **(10)** send monitoring data. Raw monitoring data is collected and stored by the VIM, and periodically pulled by the orchestrator to monitor the health and performance of CDNaaS. KPIs defined at the NSD and Virtual Network Function Descriptor (VNFD) levels can be accessed through the marketplace dashboard Monitoring.

Finally, steps **(11-12-13)** show how the vCDN Customer is able to configure CDNaaS through the Network Management System (NMS). For example, let us assume that the vCDN Customer wants to push a new content into CDNaaS. Then, he will call the appropriate primitive from the Element Management System (EMS), which will trigger a configuration event to the orchestrator through the *Vf-Vnfm-Em* interface, in turn followed by a "configuration" VNF lifecycle event that will be reached CDNaaS through the *T-Ve-VnfM* interface.

Requirement 2 ✓ Through the Marketplace GUI and the dedicated EMS interface, the vCDN customer is able to initiate and configure the service without having to manage low-level aspects of the VNF. The next section presents the evaluation part of CDNaaS.

3.5.3 Evaluation of content distribution

As the VMG inspects every HTTP transaction between the client and the server, it is important to assess, depending on the number of filters, in which extend it penalizes the throughput. To do so, an apache2 HTTP Server is deployed with bandwidth limited to 100 Mbps. The VMG is placed between the client and the server and Apache HTTP server benchmarking tool[17] is used to measure the throughput in Kbps with and without the VMG activated when 50 clients concurrently pull 1KB, 10KB, 100KB and 1MB payloads. The experiment has been reproduced 5000 times and the results are reported in Figure 3.14.

It must be noted that the impacts of the VMG for small payloads (1KB and 10KB) is very important: the original throughput of the HTTP server is reduced to 40% even if with no

[15] https://github.com/openstack/nova
[16] https://github.com/openstack/neutron
[17] https://httpd.apache.org/docs/2.4/programs/ab.html

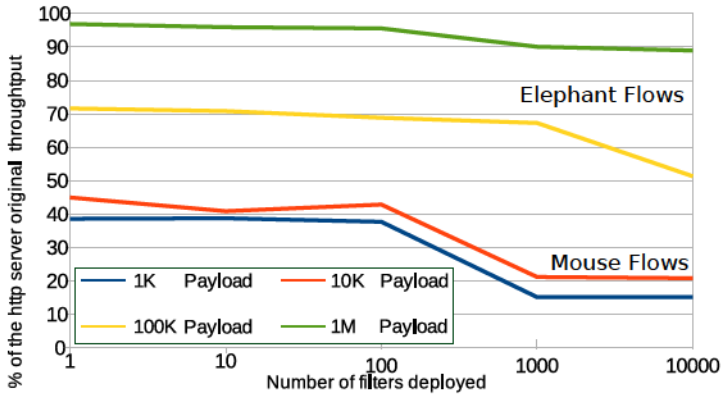


Figure 3.14: Impact of the VMG HTTP filter/action overhead on content delivery

rule is present. This rate drops even more when thousands of rules are applied to a mere 20%. Consequently, using the VMG has a huge impact for *mouse flows* which is explained by the fact that the HTTP inspection on header and URLs is done very often with this type of traffic. The proposed VMG solution is hence not suitable for dealing with very small assets, such as small images or code fragments. In that case, relying on DNS redirection would more efficient.

When the size of the payload increases, however, the throughput reduction is far less important. Indeed for 100KB payloads,the VMG can achieve 70% of line rate, and 90% for 1MB payloads. As expected, the VMG shows better results for *elephant flows*, since the HTTP header inspection routine is called less frequently for large payloads. The typical payload for files transmitted over HTTP can range from several hundreds of MBs for videos retrieved with progressive download, down to several hundreds of KBs for the typical DASH chunk. In this case, the concept of VMG implemented in software is adapted, even if there is still a lot of room for improvement.

We consider several optimization opportunities for future work. First, CPU libraries and drivers can improve packet processing time, such as DPDK [Intel, 2015]. This solution has been used successfully in a related *Virtual Traffic Classifier VNF* implemented in the T-NOVA project [Kourtis et al., 2015]. Another possibility would be to leverage Openflow extensions 1.6 to support further data-plane programming like proposed in [Bonola et al., 2017], whether in hardware or in software in the kernel space through Open vSwitch.

Another important aspect of feasibility for the VMG is its ability to scale along with the demand. The capacity of the VMG needs to be adapted dynamically, making sure that the component is not under-dimensioned and does not slow down the content delivery flow. There are two possible options: one is to provision larger VMs with more CPUs, RAM and faster disks (scale-up), the other is to create additional small VMs and increase their processing capacity by making them collaborate (scale-out). Scaling-up is easier to implement but less powerful as each VM can only grow up to the size of the underlying

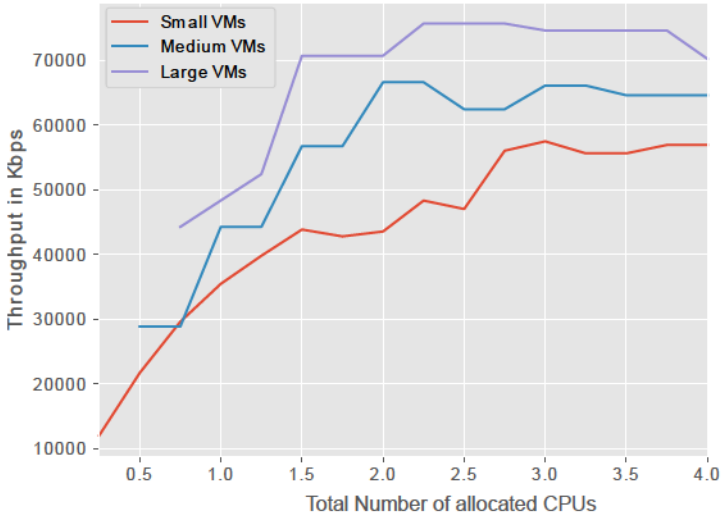


Figure 3.15: Impact of the load balancing on the VMG deployment

compute node. It is also less flexible, as the scaling-down of a VM may not be possible on the fly. Finally, contrarily to scaling-up, scaling-out has the advantage of providing high availability, since load balancers, such as HAProxy, provide mechanisms to control the availability of each back-end VM.

For this reason, a scaling-out mechanism was implemented based on several VMG instances and a HA proxy. Figure 3.15 shows the throughput achieved while increasing the number of VMs in the VMG. The experiment was carried out with different VM sizes (small with 0.25 CPU, medium with 0.50 CPU and large with 0.75 CPU, CPU resources being allocated on a 4CPU Server). We can see from the graph that increasing the number of VMs increases the throughput but the increase is not linear, and flattens between a total of 2 and 3 CPUs. After this level, the bottleneck resides on the load-balancer itself, which was limited only to 1 VM in our experiment. One way to circumvent this issue in a production environment would be to scale-out haproxy itself, and to create a hierarchy of load-balancers [Zheng et al., 2010].

The overall performance of the delivery part depends, at a large extent, on the network performances between the object storage nodes. Indeed, each content is chunked and spread over several nodes to provide redundancy and increase performance. In Figure 3.16, we used apache2's ab tool to compute the 95 percentile maximum time taken to download a 10s, 6 MB video file encoded as 600 KBps. The number of concurrent connections was increased to establish the threshold above which the video cannot be streamed at its nominal bitrate for a 5VM baseline configuration.

Two important results can be seen from the graph. First of all, there is no significant difference between the performance of Storage with or without the VMG. It means that the storage is the bottleneck in this case, and the VMG does not need to be scaled-out to

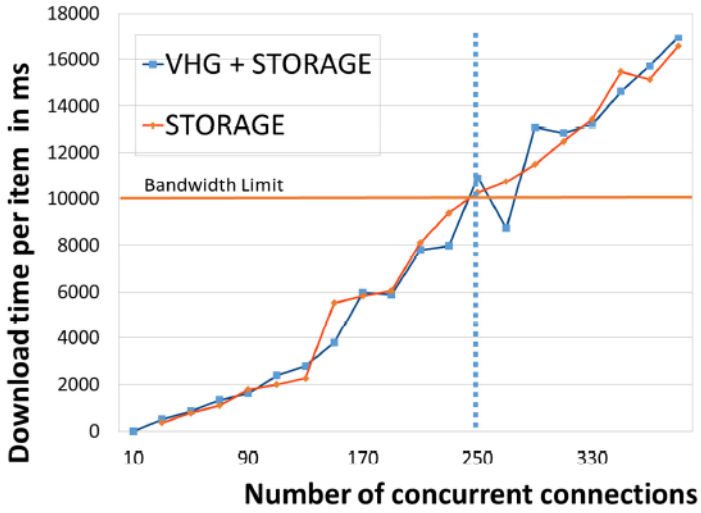


Figure 3.16: End-to-end evaluation of the content delivery components

increase performances. Next, the video can be streamed by 250 simultaneous users. This value is strongly correlated to the underlying state of the network on our infrastructure and also on the storage technology used in the platform. For example, our object storage engine is designed to use SSD disks to boost the delivery of the most used files. This feature was not available on our infrastructure and could have drastically increased performances, especially for Internet content where only a small number of items is popular whilst the rest remains unknown.

3.5.4 Evaluation of content ingestion

Ingestion of content is a very CPU and memory intensive task that can be easily scaled with the adjunction of a "worker" VM. Figure 3.17 shows a setting where the system ingests 200 videos of 20MB at an average arrival rate of 30 videos per minute. We compare the number of "pending" video jobs that are queued by the system for several settings. VNF is scaled-out, allowing the number of ingestion workers to vary. The configuration with only 1 worker does not cope with the load as it accumulates more than 120 pending videos and it depletes its video stock in more than 900s. On the opposite, the 3 VM setting manages to finish nearly on time (420s). Thanks to this design, the number of workers can be adjusted based on the characteristics of the videos and on the tolerance to delay of the customer.

Increasing the number of workers is not the only option when considering the challenge of increasing performances for the ingestion phase. Other proposals aim at using GPU virtualization in NFV environments, such as [Paglierani, 2015]. In a demo settings we

Figure 3.17: Transcoding Performances

showcased at the Mobile World Congress in 2016[18], we modified the admission workers to use an external VNF called the Virtual Transcoding Unit [Comi et al., 2016] developed by Italtel and university of Milan, and we observed a 10-fold decrease in processing time with regards to a pure software solution.

3.6 Conclusion and Future work

CDNaaS concept has been validated in a lab environment and in the real T-NOVA NFV infrastructure. All the requirements expressed in Table 3.1 were fulfilled[19]. The result is a scalable, yet modular architecture allowing a vCDN customer to buy CDNaaS to match its goals in term of quality enhancements in its content delivery network. Thanks to the deployment in operator's Points of Presence, and the fact that the network can be managed end-to-end thanks to the Wan Infrastructure Manager (WIM) network controller, ambitious targets such as maximum latency and minimum bandwidth guaranties toward the end-user premise is at reach.

The technical and design choices undertook are not the only way to address the challenges of building a vCDN-as-a-Service. CDN providers and large CPs have developed over the years numerous proprietary solutions supplementing the standard CDN features. For example, the content provisioning is addressed in a rather naive pull-based fashion, despite the fact that most CDNs benefit from smart push-based mechanisms based on content popularity models [Kryftis et al., 2016], or user-centric data mining analysis [Pallis and Vakali, 2006].

[18] https://nextnet.top/node/6

[19] a video demonstration is available online: https://www.youtube.com/watch?v=xEbs5BUN6NA

A number of approaches can be considered to increase the performances of the Virtual Media Gateway (vMG) and the content ingestion. They rely on using specific libraries dedicated to fast packet processing (such as DPDK on supported network interfaces) or cloud-base hardware acceleration (such as GPU-enabled cloud instances).

The distributed storage solution supporting the vStreamers still relies on costly high-density storage arrays in data centers; alternative technologies such as fog-scale storage could be considered to reduce cost and have an even wider geographic distribution.

Even if the implementation of CDNaaS and its integration in an NFV architecture are the first steps toward the concept of vnf-based content delivery, its integration in the current ecosystem raises several challenges:

1. Will ISPs be willing to offer third-party CDN VNF hosting?

2. What is the benefit for CDN operators and Content Providers to integrate CDNaaS in their existing deployments?

We address those questions in the next chapter by proposing a model for a mutually beneficial partnership.

CHAPTER

4 A Model for Content Delivery Collaboration: a VNF-as-a-Service Perspective

Our study in the previous chapter showed the technical feasibility of a Virtual Content Delivery Network implemented as a Virtual Network Function (VNF) deployed in a VNF-as-a-Service infrastructure, CDN-as-a-Service (CDNaaS). Even if this approach can be directly used by the Internet Service Providers (ISPs) to build their own Content Delivery Network (CDN) network to serve their own customers, its impact would be limited. Indeed, most of the traffic from Content Providers (CPs) being encrypted (e.g., Youtube claims delivering 97% of its video content with HTTPS [Schechter, 2016]), ISPs cannot reliably store it in their own cache to speed-up their customer access and save on transit costs. For this reason, we expect that the main benefit from CDNaaS would come from a collaboration between ISPs and CPs /CDN operators. When selling Virtual CDN (vCDN)-as-a-VNF, ISPs can provide a valuable service to CPs/CDN operators by offering a top-tier connectivity toward its End-Users (EUs), while saving on transit cost. In this chapter, we propose a model able to make the collaboration between ISP and CDN operators/CP possible. We detail an architecture that can leverage CDNaaS and compare its benefits with other existing collaboration schemes. We then extend this approach by proposing a user-centric collaboration that also leverages on the Customer Premise Equipment thanks to an innovative blockchain approach. We believe our proposal can represent a win-win situation for ISP and CDN /CP and has the potential to improve the quality of experience for the EU, reduce the cost for the ISP and balance fairly the added-value of content delivery.

4.1 Introduction

As we presented in Chapter 2, Real-time entertainment (video and audio streaming) is the number one service on the current Internet: it accounts for 68.90% of the Downstream Peak Period Traffic in North America. Consumers' expectations for video quality are also increas-

ing and high definition is not the only factor to improve: other aspects such as the initial time spent buffering a video as well as stalling in video has been shown to greatly impact user experience, thus engagement [Dobrian et al., 2011]. In [Nygren et al., 2010], authors identified peering point congestion, inefficient routing protocols, network unreliability and the inefficiency of existing communication protocols as factors adversely affecting such operations.

CDNs aim at solving those issues by deploying servers in strategic locations. They assign users to a close-by server, thus reducing hop count and avoiding potential congestion occurrences. However, CDNs and ISPs do not naturally collaborate. They have their own specific problems to solve resulting in conflicting outcomes. Authors in [Jiang et al., 2009] classify these problems in two categories: **Server Selection** (SS) ones and **Traffic Engineering** (TE) ones. Traditionally, SS problems are under the CDNs' responsibility. A CDN server can be chosen for a request if (1) it hosts the provisioned content, (2) the format of the content is compatible with the User Agent that performed the request and (3) the server is able to serve the content with the appropriate quality. The (1) and (2) problems are related to the core business of the CDNs and do not interfere with the ISP operations. The last problem (3), however, is tightly coupled with the way ISPs handle routing within their networks. User-server assignment choices may drastically suffer from undetected network bottlenecks or EU mis-location, despite their effort in inferring network characteristics [Krishnan et al., 2009b]. TE problems are handled by ISPs through the deployment of Internal Gateway Protocols using Open Shortest Path First (OSPF) or IS-IS. By modifying OSPF weights, ISPs can successfully avoid congested routes and implement specific routing policies. However, ISPs do not necessarily optimize their operations to minimize *end-to-end latencies*, as required for multimedia delivery.

In the pure **player CDN scenario** (Akamai, Limelight, Level3), CDN servers are located outside the ISP AS, without any direct collaboration with the ISP. When selecting a server for a particular streaming session, CDNs cannot access directly routing information (like OSPF weights) and must infer congestion either by measuring TCP packet drops or by using Explicit Congestion Notifications [Ramakrishnan et al., 2001]. This leads to inefficient allocation due to the limited view of the network [Frank et al., 2013], an increased peering traffic for each connected CDN by ISP, and a relatively high number of hops necessary to reach the CDN surrogate server.

When the ISP owns the CDN (also called **Telco CDN**), collaboration can be achieved through CDN Interconnection (CDNi) principles [van Brandenburg et al., 2015]. ISPs can exchange their footprint and capability with other CDNs, using Request Routing interfaces [Peterson et al., 2014]. Application-Layer Traffic Optimization (ALTO) [Alimi et al., 2014] information and detailed topology can be exchanged via CDNi interfaces [Seedorf et al., 2015]. However, interface specifications are still on-going, and there is no available implementation yet. This standard mechanism is challenged by "MultiCDN " providers[1] that offer a

[1] http://www.metacdn.com/what-is-a-multi-cdn

service enabling the CDN customer to rent and orchestrate several CDN providers at the same time, letting the MultiCDN managing the complexity of CDN interconnection, load balancing and building an aggregated operational report [Adhikari et al., 2012]. For example, Warpcache[2] have partnered with the biggest players (Akamai, EdgeCast, Highwinds, Hibernia, MaxCDN, Comcast, ChinaCache, G-Core, Quantil and Medianova) to offer these providers in a multi-CDN solution.

Another alternative, known as **Managed CDN**, (e.g., Akamai Aura Managed CDN [3], Google Edge Network [4] or Netflix Open Connect [5]) allows CDNs to deploy surrogate servers directly into the ISP network. This solution enables the ISP to reduce peering traffic and limits the number of hops between the EU and the content servers. Although this solution brings some cost savings to the ISP and better performance, it still offers limited integration, as the ISP is unlikely to expose its complete network map to CDNs. Scalability is also an issue since surrogate servers come as physical appliances and need to be upgraded should the traffic be increasing. Finally, an agreement must be signed between the ISP and every single CDN Provider.

A third approach consists of **Licensed CDNs** , (e.g., Akamai Aura Licensed CDN [6], StreamZilla CDN [7]), where CDN providers still deploy their servers inside the ISP infrastructure but offer more flexibility to their users. ISPs can resell CDN services to their business customers while using CDN -licensed software. Contrary to the more traditional **Telco CDN** solution where the ISP assumes the whole responsibility and expertise costs to design and implement the system and to take the role of real CDN providers, integrating and managing a Licensed CDN is much simpler.

The ultimate goal of a collaboration scheme is to improve the quality of experience of the EU while fairly balancing the benefit from the collaboration. Given the strong motivations for all the parties to collaborate, one can rightly ask why none of the collaboration schemes described earlier has been widely adopted, and the threat posed on the Internet by the increase of Over-The-Top (OTT) video consumption. There are several reasons for that. First, the content delivery market is evolving very fast and the bargaining power of each actor is changing over time, which prevent them from passing long-term agreements. Second, the evolving legislation over net neutrality, a principle which prevents ISPs from prioritizing the content of some CPs/CDNs against payment, may be an incentive to slowdown collaboration. Indeed, should net neutrality principle be abandoned, ISPs would be better of refusing any collaboration with regard to cache placement, in order to monetize the creation of fast lanes and to solve the quality problems of CP /CDN. Finally, even if ISPs are investing tremendously in network softwarization technologies, such deployments

[2] https://www.warpcache.com/
[3] https://www.akamai.com/us/en/products/network-operator/managed-cdn-solutions.jsp
[4] https://peering.google.com/infrastructure
[5] https://openconnect.netflix.com/en/
[6] https://www.akamai.com/us/en/products/network-operator/licensed-cdn-solutions.jsp
[7] http://www.jet-stream.com/licensed-cdn/

are still ongoing and not ready for production yet. Even if Cloud Providers have proven Software Defined Networking (SDN) to be beneficial when managing multi-tenant networks datacenters, the scale and complexity of deploying SDN, in an operator network, excesses the largest datacenter network.

In order to overcome the CDN -ISP collaboration challenges, we propose a Virtualized Infrastructure solution capable of matching ISP's connectivity "supply" with the CDN's connectivity "demand", towards a win-win approach. Unlike the fast-lane creation approach, where the traffic is shaped depending on the CP/CDN [Gharakheili et al., 2016], we propose to directly host the CDN functions inside the ISP network as a Virtual CDN deployed in an operator Point-of-presence (POP).

The main benefits from this approach is that it creates a new source of revenue for the ISP, installing a "two-sided market" where it can charge both EU and CP/CDN for the content delivery. With this increased revenue, the ISP can invest even more in network capacities to stay competitive, which is good for the health of the network. It also has advantages for the CP and the EUs: providing higher-definition video contents improves the quality of experience of the EUs for which the CP can apply a premium price. Finally, it also allows ISP customers to benefit from a higher quality in rural area, where the market incentive to build a full-fledged CDN network is not sufficient.

We propose several contributions to achieve these goals. First, starting from the Virtual CDN solution proposed on the previous chapter, CDNaaS, we elaborate on the deployment context of the solution. Second, we analyze, using game theoretic tools, the optimality conditions of CDNaaS compared to other collaboration schemes. Finally, we extend the model for including collaboration to all the actors of the content delivery value chain toward a user-centric delivery.

4.2 CDNaaS in ISP network: which collaboration model?

From the previous section, we saw that the most important aspects of a collaboration model between ISP and CP/CDN operators is the ability to tackle conjointly the server section and traffic engineering problems. For that, some of the control of the network must be shared, to some extent. Thanks to the virtualization of resource provided by network softwarization, this goal can be reached by (1) letting the CP/CDN Operator managing a virtual network embedded on top of the ISP physical network and (2) hosting a CDN deployment as a virtual network function.

With this aim in view we propose instantiating CDNaaS on a Network Function Virtualization platform into the ISP network distributed amongst several Network Function Virtualization Infrastructure Point of Presence (NFVI-POP)), at the edge of the network. The components that benefit the most from being located close to EUs are the cache (Virtual Streamer (vStreamer)) and the requests routers (Virtual Media Gateway (vMG)). The other CDN modules related to content ingestion can be instantiated in more remote cost-effective

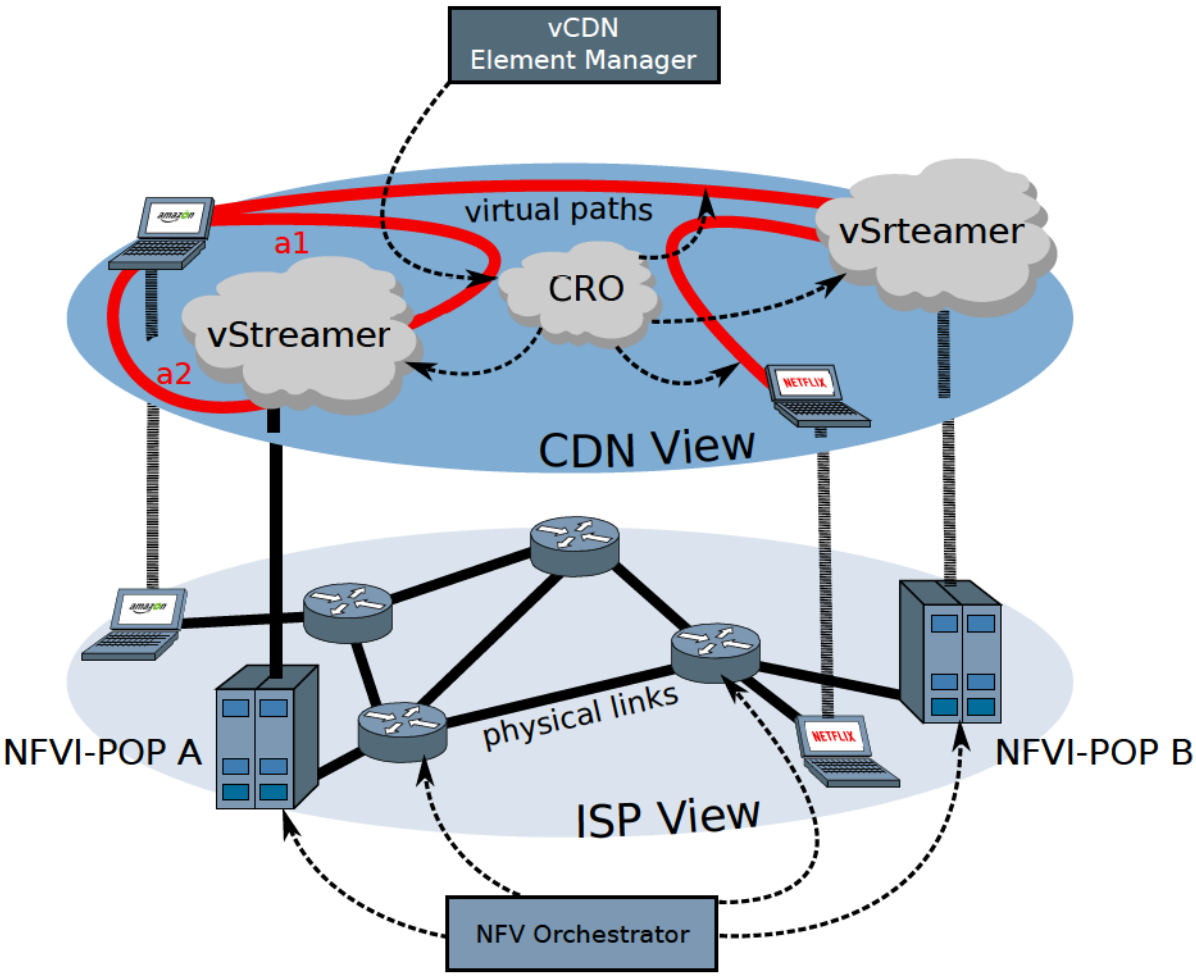


Figure 4.1: Deployement of a CDN Overlay using CDNaaS

data centers or opportunistically at edge data-center with unused spare capacity. For this reason, the rest of this section will not consider content ingestion in the collaboration scheme. We now elaborate the the mechanisms used to shared part the network control between stakeholders.

4.2.1 Collaboration architecture

As mentioned in the previous section, ISPs do not communicate details about their infrastructure, even if CDNs need some to optimize server selection. One way to circumvent this issue is, for the ISP, to present only an Overlay network to the vCDN customer, which will convey information such as end-to-end delay along the route, bandwidth reservation or congestion state. The Underlay network is not disclosed to the CDN operators, maintain-

ing the confidentiality of the ISP topology and resources, following a privacy-preserving "Connectivity as a Service" (CaaS) approach [Wang et al., 2014].

Figure 4.1 presents the overlay exposed to the CDNs, the EU domains connected to edge routers, a logically centralized Caching and Routing Orchestrator VNF and Virtual Routes. Each virtual route has specific properties and are configured by the vCDN customer to connect a certain EU domain with an NFVI-POP where a vStreamer VNF is deployed. The vCDN customer is also responsible for scaling out Streamer VNF instances in each NFVI-POP, according to its needs.

For example, if a bottleneck occurs in the $a1$ route of the Virtual Network, the vCDN customer can decide to configure the Caching and Routing Orchestrator (CRO) to offload $a1$ to $a2$ automatically. If the Streaming VNF in NFVI-POP A is the limiting factor, the vCDN customer can scale out the VNF, by requiring more VNFs to be allocated to it in the NFVI-POP A. If NFVI-POP A and routes $a1$ and $a2$ are saturated, the vCDN customer can decide to use an alternative NFVI-POP B to absorb some traffic from the EU domain by scaling out Streamer VNF, as well as assigning alternative virtual routes to EUs.

Even if the vCDN customer can interact directly on the internal behavior of its VNFs, a more flexible approach is to use the CRO to perform such tasks. In this case, the vCDN Customer can define a high-level policy and deploy it to the CRO, which in turn would be responsible for implementing the low-level details.

4.2.2 ISP platform key components for collaboration

Using a software platform to perform network operations allows a great deal of flexibility and innovative network management that is not possible using physical appliances. Costs are kept low by running out of commodity hardware. That is the reason why the deployment of services inside the ISP network over an Network Function Virtualization (NFV) architecture is envisioned. Besides the vStreamer VNF and the vMG, two other necessary modules to enable a fully mutually beneficial ISP -CDN collaboration are envisaged. The first interface is business-oriented: *the Marketplace* allows deploying CDNs VNFs to the ISP and negotiating a Service Level Agreement (SLA) between them. SLA elements can be network-related (guaranteed bandwidth availability, maximum e2e delay...) or system-related (reserved Central Processing Unit (CPU) resources, disk space...) The other interface is technical: the ISP 's *vCDN Overlay Network Controller* exposes an Application Programming Interface (API) to realize virtual route selection.

4.2.2.1 The Marketplace

The Marketplace (see section2.3.4) is a business-oriented module allowing (1) to upload the Virtual Deployment Unit (VDU) of CDNaaS to the ISP NFV platform and (2) to negotiate a service SLA between the ISP and the CDN, as presented in Figure 4.2.

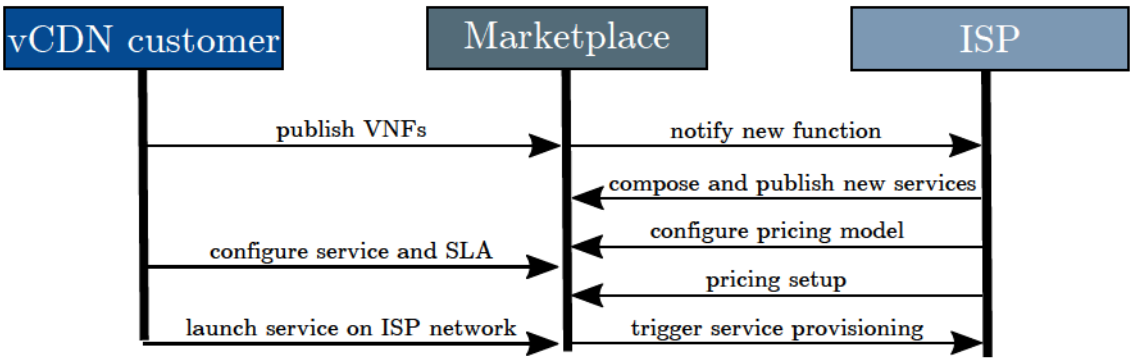


Figure 4.2: CDN, ISP and Marketplace interactions

First of all, vCDN customers upload their VNFs to the Marketplace triggering a notification event on the ISP side. The ISP builds a service composed of those VNFs. This new service is published on the marketplace's service catalog and is made available to the CDN Providers for configuration and launching. During service configuration, the vCDN customer expresses its requirements in an SLA, as already exemplified in [Famaey et al., 2012a] for multimedia service. The SLA contains system requirements[8] and network resources required to run the service. After this step, the vCDN Customer can accept the contract and launch the service.

Let us now give a quick example of SLA that can be established between an ISP and a vCDN customer, using notations from Figure 4.2. In this example, the vCDN customer has very bad performances for a particular domain $d1$ within the ISP network due to missing peering agreement for this area. The possible SLA could be: serving 200 simultaneous connections to $d1$, average bitrate of 4 Mbps of downstream traffic. The vCDN customer has documented, in the Marketplace, that its Streamer VNF can support up to 50 concurrent connections on a **m1.large**[9] VM flavor. NFVI-POP A being the closest from $d1$, the ISP dedicates there up to 4 m1.large VMs. Then, the ISP sets-up a virtual route $a1$ with $200 * 4Mbps = 800Mbps$ allocated bandwidth, making sure that the maximal RTT from the user

[8] Two directions can be taken when specifying system requirements, depending on the type of integration between the vCDN Customer and the ISP.

- They can be low-level system requirements, such as the number of virtual CPUs, amount of RAM and storage. In this case, the vCDN customer can mix and match any type of system metric. This makes sense for its own implementation, as this is done for private clouds.
- They can also be simplified by summarizing the need of a particular VNF in term of number of VMs with a particular flavor, as it is the case for most public Clouds that propose "varying combinations of CPU, memory, storage, and networking capacity to choose the appropriate mix of resources for an application" [AWS, 2017].

[9] default m1.large profile for OpenStack is 8GB of RAM , 80GB of disk and 4 Virtual CPUs (vCPUs)

to the POP is not greater than $524288b/4,000,000bps = 131ms$ (using the maximum 524288 bits TCP window size).

4.2.2.2 vCDN Overlay Network Controller API

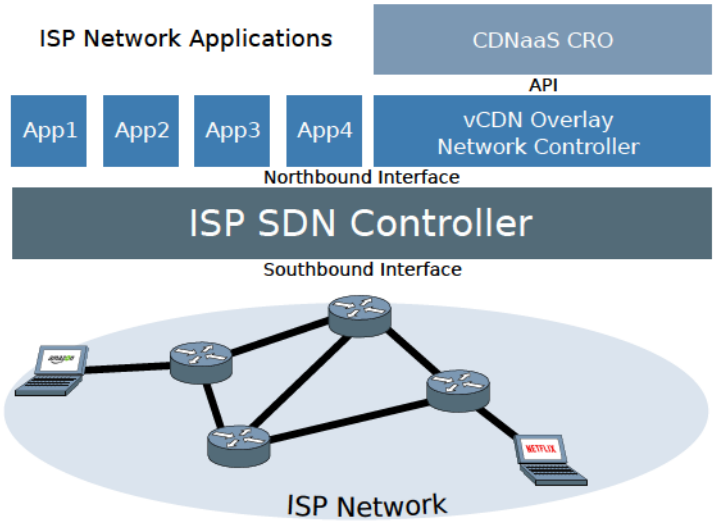


Figure 4.3: The vCDN Customer programs CDNaaS' CRO to orchestrate both internal resources (VMG and vSTR) and the vCDN Overlay Network Controller

In [Aflatoonian et al., 2015] author promote, using SDN, to build a platform to accommodate thirds party control applications. Their works do not necessarily target a specific use-case, however, they describe how external "Guest Controllers" can be used to access monitoring information and alter the control platform through a new network abstraction layer (Figure 4.3). Tenants are provided APIs with abstract views and pre-defined levels of control over their dedicated virtual networks, with no concerning about the underlying type and number of controllers, as well as topology of physical networks [Jiang et al., 2016].

In our case, vCDN customers leverage this approach through the CRO which can react and deploy content delivery policy updates when the states of the virtual route require it. An example of such a policy is provided in section 4.2.4.

4.2.3 Comparison of the collaboration models

The following section compares the different collaboration models presented before with a CDNaaS approach. We start by expressing formally the ISPs and CDNs collaboration schemes, we then use a game theoretic approach to study the conditions for which CDNaaS offer the lowest cost.

4.2.3.1 Study of CDN and ISP profitability

We established common formal notations to model ISP and CDN profitability, reported in Table 4.1. We then use these notations over all the collaboration scenarios to establish the associated costs and earnings in each case. Each model is then used to fill the profitability matrix 4.2 so they can be easily compared.

Table 4.1: Notations and estimates for use-case 1

	Units	Description	Estimates	Source used for estimates
b	$	internal bandwidth costs	-	-
t	$	transit costs	$21,000$	bandwidth cost[10], Internet daily traffic at POP [11]
P	$	profits margin of CDN	-	-
α	%	share of the traffic toward CDN device placed inside ISP 's Autonomous System (AS)	80%	Akamai [12]
β	%	share of the traffic toward ISP Business Customers	10%	-
s	$	cost of ISP NFV infrastructure	80,000$	hardware requirements for swiftstack storage, based on commodity prices with a 3 year asset depreciation
l	$	licenses paid by ISP to CDN	24,000$	based on Swiftstack pricing[13]
r	$	CDN business margin	20,000$	with median price 0.025$ per GB
d	$	cost for designing and maintaining CDN software	555000$	acquisition of a CDN Startup of a 20M$ valuation
$f(.)$	$	pricing function used by the ISP to bill the CDN	-	-
$g(.)$	$	pricing function used by the CDN to bill its customer	-	-
$g^*(.)$	$	current market pricing function used by the CDN to bill its customer for premium service	100,000$	cdn77 pricing website

[10] http://drpeering.net
[11] https://www.franceix.net/en/technical/traffic-statistics/
[12] https://community.akamai.com/community/web-performance/blog/2015/03/16/what-is-a-good-origin-offload-number
[13] https://www.swiftstack.com/pricing/

Table 4.2: Profitability matrix containing Earnings and Costs for CDN and ISP for each collaboration scenario

	ISP Earnings	**ISP Costs**
1 - No Collaboration	0	$b+t$
2 - Managed CDN	0	$(1-\alpha)(b+t)$
3 - Licensed CDN	βr	$(1-\beta)(b+t)+\beta(s+l)$
4 - Telco CDN	βr	$(1-\beta)(b+t)+\beta(s+d)$
5 - CDN -as-a-VNF	$f(\alpha,\beta)+\beta r$	$(1-\alpha-\beta)(b+t)+(\alpha+\beta)s$
	CDN Earnings	**CDN Costs**
1 - No Collaboration	P	t
2 - Managed CDN	P	$(1-\alpha)t$
3 - Licensed CDN	$P+\beta l$	$(1-\beta)t$
4 - Telco CDN	P	$(1-\beta)t$
5 - CDN -as-a-VNF	$P+g(\alpha,\beta)$	$(1-\alpha-\beta)t+f(\alpha,\beta)$

4.2.3.1.1 No Collaboration

In this baseline scenario, ISPs have to assume a cost b of internal bandwidth to accommodate the multimedia streams from EUs to the server selected by the CDNs for serving the content. ISPs and CDNs must pay a transit cost t to exchange data between their respective AS. This cost could be paid to transit providers (typically a Tier-1 ISP) or can come from paid peering or possibly from settlement-free peering (in which case some collocation costs can occur). We consider that the CDNs generate P net profit from selling their service to CPs (discounting every cost except t). ISPs however make no profit in this scenario, as all the added value goes to the OTT Provider.

4.2.3.1.2 Managed CDN

In this model, the CDN redirects a fraction $\alpha \in [0;1]$ of its traffic to an appliance placed within the ISP's network. Both CDN and ISP win, as they need to cover only a fraction of the external traffic's cost $(1-\alpha)$ as the flows stay within the ISP AS and do not need to enter the CDN AS.

4.2.3.1.3 Licensed CDN

In this scenario, the ISP resells CDN services to its business customers. To run the service, ISPs need to maintain an infrastructure. Here, ISPs do not own the software, they pay a license fee to the CDNs. The pricing l corresponds to the price that the ISP would have paid if 100% of the traffic had been served by the software under license and likewise, s is the price that the ISP would have covered if 100% of the infrastructure had been used. For the sake of simplicity, we assume that those costs are linear with respect to the share of traffic served by the system, noted β. We also assume that the ISPs pay only for the share of

infrastructure used. This assumption holds if the ISP's "CDN business unit" performs an internal re-billing for infrastructure costs or if the ISP buys computing power from a cloud provider. So, in this case, the ISPs only covers βl licensing fees and only βs infrastructure cost.

The important advantage of such collaboration is the reduction of cross-AS traffic by a ratio $(1-\beta)$ and the extra earnings r for the ISP, which is also considered linear wrt to β.

4.2.3.1.4 TelcoCDN

This case is similar to the previous one, except that instead of licensing costs, ISPs own the software and assume the costs of engineering and maintenance d. No payment is made to the CDNs.

4.2.3.1.5 CDN-as-a-VNF

This scenario combines the advantages of Managed CDN and Licensed CDN with respect to external traffic reduction, as both α and β factors are discounted. ISPs still have to support infrastructure costs s but also benefit from service resell. Their role is somewhat similar to Cloud Provider's role, except they have more leeway to optimize the network end-to-end, from the servers to the EUs. Their pricing function noted f is used to determine the price the CDNs will pay for the hosting. The CDNs have their own pricing function g which will be used to bill their own customers for this premium service.

4.2.3.2 Optimality conditions for the CDN-as-a-VNF strategy

This section deals with the formalization of the optimality problem, considering Game Theory principles. First, some definitions are outlined.

Definition 1. CDN-ISP collaboration is a Game noted G_{collab} for which each CDN's (resp. ISP's) strategy $s_i^C \in S^C$ with $i \in [0, |S^C|] = I^C$ (resp $s_j^I \in S^I$ with $j \in [0, |S^I|] = I^I$) has payoffs p_i^C (resp. p_j^I) obtained by subtracting the corresponding costs and earnings from Table 4.2.

Definition 2. The CDN-ISP collaboration model χ is strictly optimal if the corresponding combined strategies $(s_\chi^C, s_\chi^I) \in S^C \times S^I$ form a pure strategy equilibrium for G_{collab}.

This means that the problem can be addressed by solving a set on inequalities arising from the fact that s_χ^C and s_χ^I are strictly dominant strategies.

$$\left(\forall s \in S^C / \{s_\chi^C\}, s \ll s_\chi^C\right) \Leftrightarrow \left(\forall i \in I^C, p_i^C \geq p_\chi^C \Leftrightarrow i = \chi\right)$$

$$\left(\forall s \in S^I / \{s_\chi^I\}, s \ll s_\chi^I\right) \Leftrightarrow \left(\forall j \in I^I, p_j^I \geq p_\chi^I \Leftrightarrow j = \chi\right)$$

Definition 2 is restrictive and may not apply for all the cases. However, it provides a simple tool to study the most important factors that condition the success of the CDN-as-a-VNF strategy. We plan to study, as future work, more permissive forms of optimality,

where different strategies can be chosen for different use cases. Let us now determine the parameter constraints enabling strict optimality of the CDN-as-a-VNF strategy. First, we analyze the existing strategies dominations for the ISP and for the CDN.

4.2.3.2.1 For the ISP

The uncollaborative strategy is dominated by the Managed CDN strategy, as $\alpha > 0$ so we have $s_1^I \ll s_2^I$. We are left with the system: $s_2^I \ll s_5^I$, $s_3^I \ll s_5^I$ and $s_4^I \ll s_5^I$:

$$\begin{aligned} f(\alpha,\beta) &> \alpha s + \beta(s-r) \\ f(\alpha,\beta) &> \alpha s + \beta(s+l) \\ f(\alpha,\beta) &> \alpha s + \beta(s+d) \end{aligned} \tag{4.1}$$

4.2.3.2.2 For the CDN

Like in the ISP case, the uncollaborative strategy for the CDN is dominated by every other strategy as $\alpha > 0$ and $\beta > 0$ since there will always be some traffic toward the CDN server located within the ISP AS. So we have $s_1^C \ll s_i^C, \forall\ i \in \{2,3,4\}$. As well, we can also see that the Licensed CDN strategy will always dominate the Telco CDN strategy, as $\beta l > 0 :\ s_4^C \ll s_3^C$. Finally, the only two inequalities to consider derive from $s_2^C \ll s_5^C$ and $s_3^C \ll s_5^C$

$$\begin{aligned} g(\alpha,\beta) &> f(\alpha,\beta) + (\alpha-\beta)t \\ g(\alpha,\beta) &> f(\alpha,\beta) + (l-t)\beta \end{aligned} \tag{4.2}$$

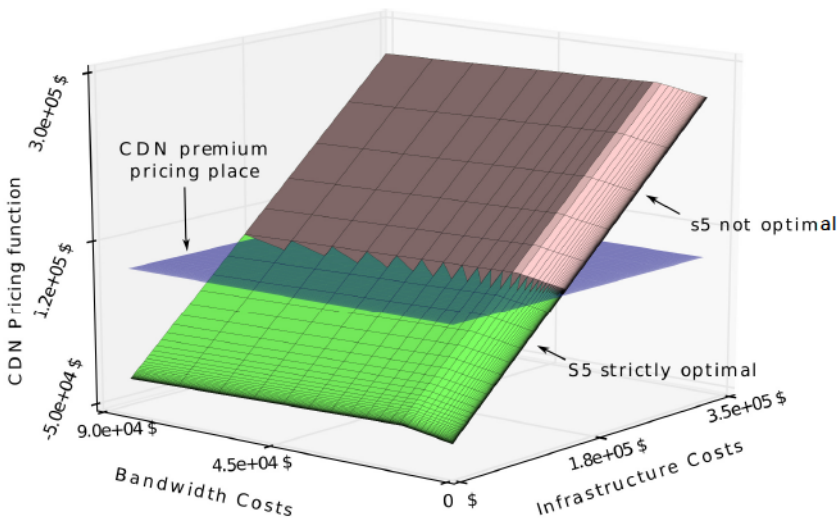


Figure 4.4: Optimality zone for the CDN-as-a-VNF strategy (5)

On top of the optimality conditions exposed in equation (4.2), we add another inequality based on current market price for premium[14] CDN delivery.

$$g(\alpha, \beta) \leq g^*(\alpha, \beta) \tag{4.3}$$

If ISPs pricing is above the existing market price for premium CDN, then the proposed model is not feasible as it would be outperformed by the currently deployed one. Let us now evaluate the principles of the costs estimation in the proposed CDN-as-a-VNF model.

4.2.3.2.3 Costs estimation

we considered a use case *UC1* where:

- A CDN delivers on average 45GB per user per month. Those numbers correspond to the average data consumption of a Netflix User[15].
- A small ISP in France handles 20% of the overall traffic of this CDN (1,000,000 paying Netflix users in France).

To calculate t, the bandwidth needed to support this load corresponds to the average hourly peak bandwidth roughly amounting to 5.63%. With a bandwidth priced at 0.63$ per Mbps/month, it sums up to $4.10^{10} \times 200000/(30 \times 0.0563/(60 \times 60)/(0.63 * 8.10^6) \times \approx 21,000\$$. The cost of the ISP infrastructure has been tailored to support the entire Netflix catalog (3PB) over 6 POPs. We followed SwiftStack (an open source object storage software that can be used in a CDN use case) hardware specifications[16] and added housing costs[17] and maintenance cost handled by third party[18]. Concerning the hardware investment, in order to determine the monthly cost, we made the assumption that it would be amortized linearly in 36 months. To estimate licensing fees l of the CDN, we used public software license prices from SwiftStack. We estimated the price of building a CDN software d by assuming that the ISP would acquire a small CDN company[19] for 20M$ and amortized it over 36 months. All these estimations are summarized in Table 4.1, along with the pointers to data.

Considering those estimations, the evaluation conducted towards the performance of our CDN-as-a-VNF model is depicted in Figure 4.4. We make ISP infrastructure cost s (resp. ISP bandwidth costs t) vary from $4s$ to $s/1000$ (resp. $4t$ to $t/1000$). The flat surface represents the market price g^* that the CDN currently charges for "Premium" CDN delivery. The oblique surface represents the break-even values for the CDN pricing function g. This surface has been colorized in green to emphasize the values where the g function is below

[14] For premium CDN, more POP are used for content replication, increasing the average quality of service and price

[15] TDG : Netflix Streaming Volume - http://tdgresearch.com/

[16] SwiftStack : Hardware Reference Architectures - https://swiftstack.com

[17] Housing Pricing - https://www.infomaniak.ch/en/housing

[18] Canonical Bootstack - http://www.ubuntu.com/cloud/openstack

[19] e.g. Telstra bought video platform company Ooyala - http://reuters.com

the premium CDN fare. It denotes the area where our proposal outperforms the current premium CDN pricing model. The zone where the surface is colorized in red corresponds to a zone where the optimality for CDN as a VNF is not feasible. This occurs for high infrastructure costs, where the ISP cannot cover the infrastructure cost without making the CDN billing a price above current market price g^*. Let us now evaluate quantitatively the possible gain of the CDN-as-a-VNF approach in the case of UC1, following the assumptions made noted θ, and let us discuss on the potential cases of revenue sharing.

4.2.3.2.4 Balancing the revenue

Considering UC1, the estimated gain of the proposed CDN-as-a-VNF approach is $g^*(\theta) - g(\theta) \approx 40,000\$$, corresponding to a 60% increased profit. When this approach is used, three possible cases can occur:

- ISP stays at break-even and the CDN can increase its pricing g up to the premium CDN market price g^*
- the CDN stays at break-even and the ISP can increase its pricing f until $g(.) = g^*$
- benefits for the approach $(g^* - g(.))$ can be shared between ISP and CDN.

Another interesting feature with this approach is to let market forces decide how to split the benefits of collaboration. By negotiating the balance of revenues in the feasible value space, both ISP and CDN are able to take into account their respective bargaining power to reach a fair balance. For instance, if the CDN does not have a good peering for a certain zone, it can reduce its profits to make sure that the ISP accepts its collaboration request. On the other hand, if the ISP has a lot of spare resources, it can reduce its margin to encourage the CDN to use its infrastructure.

4.2.3.3 Qualitative comparison of collaboration models

On top of the quantitative advantages shown earlier, the CDNaaS approach has other qualitative aspects worth noting. Table 4.3 sums up the main aspects for CDN-ISP collaboration over the different models according to the following set of features:

4.2.3.4 ISP-CDN Collaboration

The lack of collaboration between ISPs and CDN providers is as technical as it is business related. The only truly non cooperative case is the Pure Player CDN model where all traffic is exchanged between ISPs and CDNs ASs. By providing a technical way to collaborate fostered by business incentives, our solution allows matching the demand and the supply on the data delivery market, for the benefit of the end-user.

	Favor Collaboration	Avoid Lock-in	Scalable
Uncollaborative, Pure Player CDN	- -	+ +	+
Telco CDN	+ +	+ +	+
Managed CDN	+	+	- -
Licensed CDN	+ +	- -	-
CDN -as-a-VNF	**+ +**	**+ +**	**+ +**

Table 4.3: Comparison of different CDN deployment models

4.2.3.5 Lock-in

The possibility to accommodate and create competition between several actors over a given service is a good argument in favor of CDN-as-a-VNF. Licensed CDN, on the contrary, binds durably the ISP to the CDN.

4.2.3.6 Scalability

Scalability is directly achieved when the actor is in full control of its platform like in the uncollaborative case or the Telco CDN one. Traditionally, both Managed and Licensed CDN solutions relying on physical appliances are not scalable enough to respond to the trend of increasing and dynamic traffic. The VNF approach is the best option since automatic scaling-out is natively supported on the platform.

4.2.4 CDNaaS Management Evaluation

This section evaluates the capacity for the vCDN customer to manage CDNaaS through an example of an advanced content delivery policy programmed in the CRO. CDNaaS can be configured to implement fine-grained delivery policy, leveraging both virtual routes between the EUs and CDNaaS cache for enhanced content distribution and legacy best effort routes between the EUs and an origin server managed by the vCDN customer, located outside the ISP Network. In our experiment, the vCDN customer aims at implementing the following policy:

- If an EU terminal does not support High Definition (SD user), the EU gets its content from the origin server through best effort.
- If an EU terminal supports HD (HD EU), then two cases can occur:
  - if the virtual route is not saturated, then each HD EU will get its content from CDNaaS with HD quality.

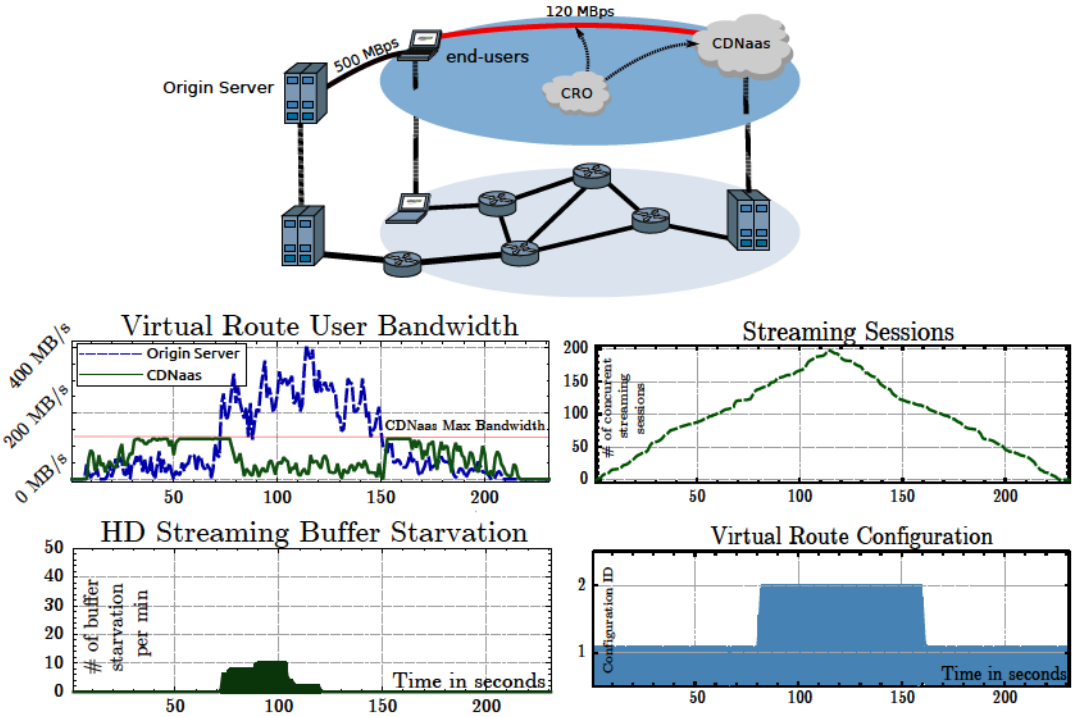


Figure 4.5: Online CDNaaS management results.

– Otherwise, if the virtual route cannot accommodate all the traffic, (1) *basic EU* fetch their content from the origin server (and get SD Quality) while (2) *premium EUs* still use CDNaaS and can enjoy HD quality.

This use-case reflects the tendency of providing commercial offerings that allow some users to access higher quality content by paying a premium[20].

We have generated 200 Video on Demand (VOD) HTTP adaptive streaming sessions with an average arrival rates of 2/s. Users are separated in two classes: first, the SD users (50% of the total) having a target bitrate of 100KBps and then the HD users with a 200 KBps target bitrate. HD users are further divided into two classes, *i.e.*, premium users (25% of the total) and the basic users. To differentiate SD and HD premium and HD basic users, the vCDN customer can configure the CRO to deploy corresponding filters and actions on the vMG (See Section 3.3.2.0.1). The separation between premium and basic users uses HTTP cookies while the SD/HD distinction is made by analyzing the HTTP user-agent header field. Implementing such policies is made possible thanks to the fine-grained management of HTTP messages by the vMG ; in this case, Domain Name Server (DNS)-redirection techniques would have been too coarse-grained.

[20] As an example, Netflix charges 2$ per month for 4K contents

The vCDN customer has created two different configurations: in the first one, every HD session is routed to the CDNaaS, since it provides guaranteed Quality of Service (QoS). SD sessions are still routed to the origin server outside the ISP AS, delivering content in best effort. In the second configuration, only premium users are served by CDNaaS. This second configuration is activated automatically whenever the bandwidth used on the virtual route is greater than 80% of the available bandwidth. The first configuration is reactivated when the virtual route bandwidth goes below 50% of the allocated bandwidth. To avoid oscillations between configuration 1 and 2, we used a rolling average window of 25s. Once configured through the Element Manager, the policy is loaded in the CRO, which in turn configures the virtual routes.

The results of this experiment are shown in Figure 4.5. It can be seen that the virtual route link starts to reach its limit at 50s. The CRO detects this saturation at 75s (due to the rolling average), and installs configuration 2. At this point, the basic users HD traffic is routed to the origin server, causing a drop in the virtual route traffic and an increase in the origin server link. Thanks to the responsiveness of the CRO mechanism, only a few buffer starvations are reported at the very beginning of the traffic burst (buffer starvation causes playback interruption in the video, which degrades Quality of Experience (QoE)). Once the quantity of traffic on the CDNaaS decreases, configuration 1 is reactivated, enabling every HD user to benefit from HD quality.

4.2.5 Discussion on Net Neutrality

Telecom regulators have imposed the principle of net neutrality to ISPs stating that all data on the Internet must be treated the same without discriminating or charging users differentially. It is of paramount importance for free innovation and is considered by some as a Human Rights [Belli, 2016]. In this section, we discuss how CDNaaS concept relates to net neutrality.

When reviewing the guidelines published by the European Regulation agency BEREC [BEREC, 2016] regarding the interpretation of EU Regulation 2015/2120 laying down measures concerning open Internet access, the VNFs used exclusively by ISPs to run from their NFVI-POPs are excluded from net neutrality regulations as per Article 3(5) if their purpose is to support specific application or services other that internet access: "*shall be free to offer services other than internet access services which are optimised for specific content, applications or services, or a combination thereof, where the optimisation is necessary in order to meet requirements of the content, applications or services for a specific level of quality.*"

Moreover, Article 3(3) Recital 9 states that "*The requirement for traffic management measures (...) not preclude providers of internet access services from implementing (...) traffic management measures which differentiate between objectively different categories of traffic (...) in order to optimise overall quality and user experience, (...) permitted only on the basis of objectively different technical quality of service requirements (for example, in terms of latency,*

jitter, packet loss, and bandwidth) of the specific categories of traffic, and not on the basis of commercial considerations." Based on this regulation, we conclude that traffic shaping for VOD service which have specific requirements in term of latency to reach very high quality is possible given that no differentiation is made between CPs hosted in POPs.

This observation must be weighed against the fact that even if no CP hosting its VNFs in the operator network is privileged in regard to another CP hosting its VNFs, they benefit from an have an advantage compared with CPs that would not. In fact, this situation already happens today when the ISP rents a rack to the CP/CDN for installing its cache or when the ISP operates a TelcoCDN. In the case where the ISPs restrict the access to their POPs through commercial advantage toward large CPs, this would in turn unfairly advantage them. Should the virtual CDN hosting solution be deployed, regulation must be created to enforce a fair access to the service.

We plan to conduct future work with legal researchers to study these aspects, which may be of particular interest given the evolution of legal framework regarding net neutrality in the US [of Chairman Pai, 2017].

4.2.6 Summary and limits of the proposed approach

We proposed the design for an ISP NFV platform in which CDNs can run their delivery functions by crafted technical and business solutions addressing shortcomings of current collaboration models. ISP-vCDN customer collaboration problem was modeled as a Game and the optimality conditions were investigated using sensible estimates. The approach was evaluated by deploying our software on a virtual testbed to implement content distribution policies that would not be feasible through the actual CDN deployment schemes.

The strengths of CDNaaS are also its limits. It provides added value by improving the Quality of Experience and by rationalizing network management through collaboration, but this collaboration is performed between two actors that tend to lock-in EUs from using alternative services:

- ISPs have a de-facto monopole in providing vCDN services to CPs, since EUs multi-homing is very rare. In other words, if a CP wants to provide good quality to EU through vCDN, it needs to agree with its ISP.
- Large CPs tend to be the one-shop stop for every OTT services, meaning that it is difficult for small and independent CPs to enter the market. EUs suffer from limited options, driving up the prices and limiting the access to alternative cultural goods.

To tackle this issue, the next section presents a model that promotes competition within the content delivery market while assuring that the network and the EUs are still gaining from the collaboration between actors.

4.3 A User-Centric Collaboration Model

In this section, we propose a user-centric approach that helps the necessary reshaping of the content delivery ecosystem. We study how blockchain-powered smart contracts and network service chaining can be exploited to support such novel collaboration schemes. Finally, our findings suggest that the proposed solution can complement existing technologies by supporting a wide range of business cases while, at the same time, significantly reducing costs.

4.3.1 The content delivery ecosystem

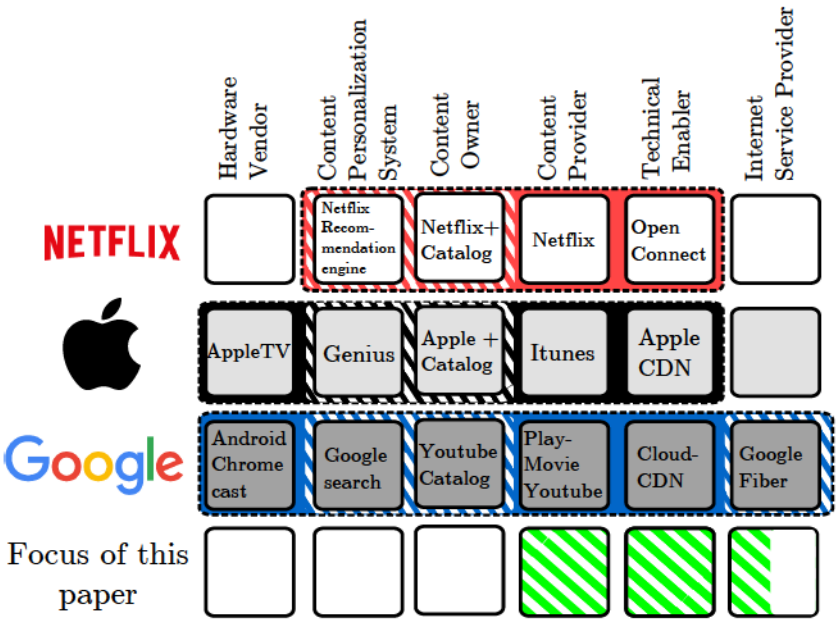


Figure 4.6: Competition Loci in OTT content delivery.

Confronted with the challenges of delivering high-quality content to an ever growing number of users, a new type of architecture started to emerge [da Silva Gonçalves, 2016]. This layered delivery architecture promotes a clear separation between:

1. Hardware vendors;
2. Content personalization systems;
3. Content Owners;
4. CPs;

5. Technical Enablers (TE);
6. ISPs;
7. EUs.

In the near future, this model will be strongly challenged, given the current trends toward vertically integrated services. For example, Netflix stopped using third party content delivery network providers, relying exclusively on its own Open Connect system, making a single company responsible for recommending, selling, producing, owning, and delivering content [Böttger et al., 2016].

In [Chuang, 2011], Chuang advocates for future Internet architectures to be "designed for competition", as a mean to achieve greater health and sustainability for the network. The main factor toward ensuring such a design is to *permit different players to express their preferences for a service delivered by various providers.* Following this nomenclature, we identify, in Figure 4.6, the six loci of competition of the content delivery market. Businesses often span over several competition loci, leaning toward more vertically integrated services. Controlling a certain locus has repercussions on others. For example, an EU cannot choose an alternative TE once he has chosen a CP. Our approach is focused on the most competition-challenged ones.

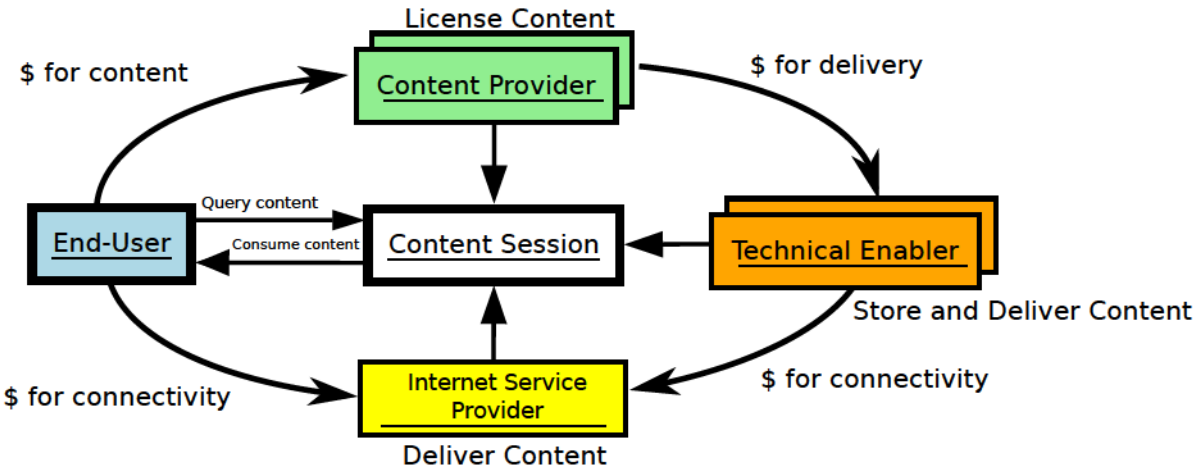


Figure 4.7: Stakeholders interactions in the content session.

To represent the dynamics behind content delivery, Figure 4.7 shows the functional interactions between stakeholders. First, the EU initiates a content query; then the CP, TE, and ISP collaborate to run a "content session" representing the actual consumption of the media by the EU. This schema highlights the current status quo in content delivery, but at the same time, it can also serve as the starting point of a more competitive ecosystem design, where:

- The EU expresses his desire to watch a specific content, along with QoE specifications (e.g., minimal video resolution) in an enriched content query.
- Several CPs read the query and respond with a content offer.
- Several TEs offer their collaboration on the content delivery session, each relying on different technologies and network configurations.

The best content session should be dynamically negotiated between actors providing the desired QoE at the lowest cost. The cost can be broken down into three parts:

1. The licensing cost charged by the CP to provide access to the content;
2. The delivery cost charged by the TE for hosting and delivering the content;
3. The network cost charged by the ISP to the TE to transfer the content to the EU.

Implementing a trusted, scalable platform able to handle negotiation messages from different stakeholders and process them according to specific business rules can be highly challenging. In this context, the blockchain is perceived as an efficient novel software architecture building block that allows reaching a distributed consensus for transactional data without the need for a trusted centralized party [Xu et al., 2016] [Xu et al., 2017]. It consists of a read and append-only distributed database that maintains a list of records, called blocks, secured from tampering and revision as each block contains a timestamp and a link to the previous block. Blockchain offers the assurance that data cannot be modified retroactively once recorded. A decentralized consensus can be achieved using specific algorithms such as proof-of-work, proof-of-stake, or Practical Byzantine Fault Tolerant (PBFT) [Castro et al., 1999]). Blockchains can be used in a wide variety of use cases, such as monetary transactions like Bitcoin [Nakamoto, 2008], medical records [Zouarhi, 2017], and even network control [Bozic et al., 2016].

Here, we propose a model for collaborative blockchain-based video delivery. First, a decentralized brokering mechanism is introduced to create content sessions through the collaboration of a CP and a TE. Second, dynamic service chains are exploited in order to benefit from link diversity of different TEs, including traditional CDN operators, virtualized CDNs following the CDNaaS approach and user-centric resources. This approach extends the previously presented collaboration mechanism by including the role CP not only as the customer of the CDNaaS solution, but also as an actor that licenses the content to be distributed. As such, it faces other competing CPs that try to license the same content to EUs.

4.3.2 A Model for Collaborative Video Delivery Based on Blockchain and Network Functions Virtualization Concepts

In this section, we describe a model using a blockchain to implement a decentralized brokering mechanism enabling CPs and a Technical Enablerss (TEs) to compete and collaborate for the instantiation of the best content delivery session. The blockchain is also used to monitor the state and quality of the content delivery. Finally, we elaborate on an mechanism used by the CPs to reward TEs for storing their content, improving their distribution throughout the network.

4.3.2.1 A Blockchain-Based Content Delivery Management Mechanism

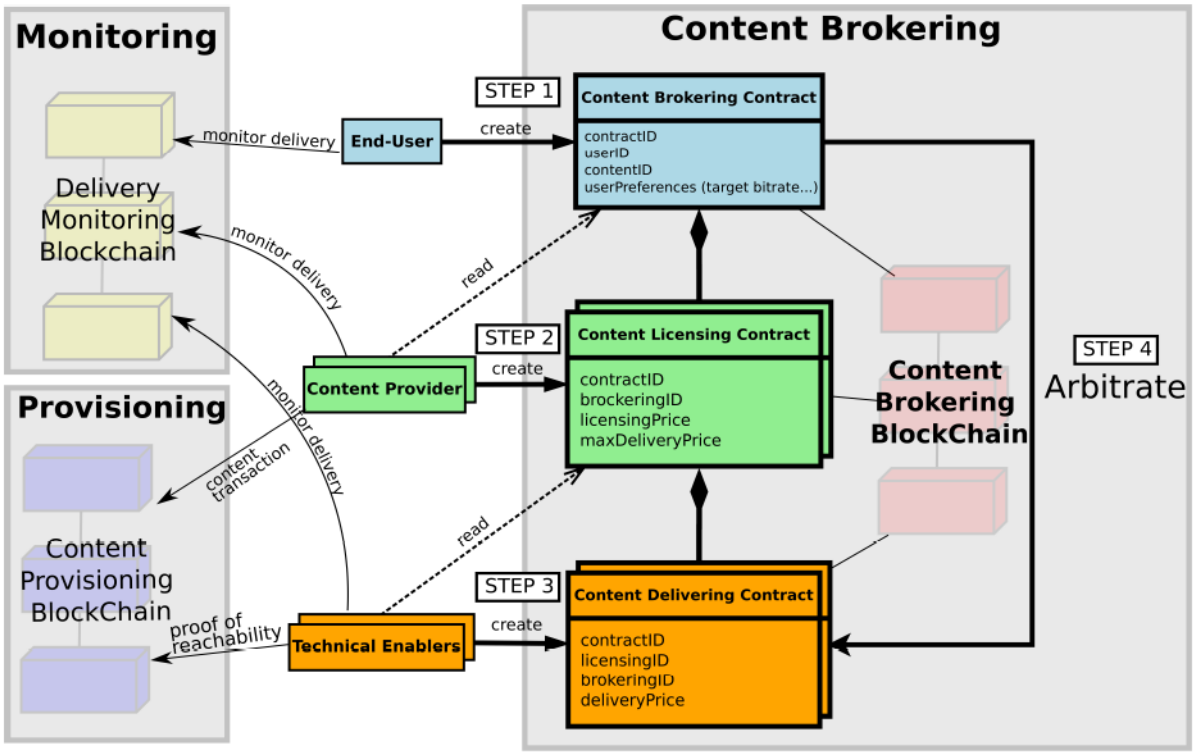


Figure 4.8: Blockchain-based model for collaborative video delivery.

4.3.2.1.1 From Blockchain to Smart Contracts

Current popular implementations of blockchains, such as the one supporting Bitcoin, have been successful at handling simple monetary transactions. However, the lack of native support for advanced programmability encouraged the development of a new generation of blockchain, extending the semantics of transaction through "smart contracts" [Szabo, 1997].

Written in a Turing-complete language, smart contracts can process data on-chain to implement complex business rules. They can be useful in automating business processes in a trusted way, by allowing all stakeholders to process and validate contractual rules as a group [Hull et al., 2016].

4.3.2.1.2 Implementing Content Delivery Processes with Smart Contracts

We envision a content delivery brokering mechanism as a series of small smart contracts. Each contract has a unique identifier and some data fields, and can perform actions such as creating a new contract or updating the state of the blockchain. Contracts actions are triggered off by on-chain data update (i.e., creation of a new contract) or time. The proposed model, as shown in Figure 4.8, is composed of several blockchains, each one implementing a specific feature used for content distribution, as follows:

- The **content brokering blockchain** handles the negotiation of the content delivery session. EUs, CPs, and TEs publish smart contracts that will be used to determine the best mix for the session.
- The **delivery monitoring blockchain** collects and processes proofs of fulfillment of the delivery contract.
- The **provisioning blockchain** is used by CPs to handle the diffusion of contents on the TE's storage devices.

4.3.2.1.3 Content Delivery Session Brokering

Once the query arrives in the blockchain, a Content Brokering Contract (CBC) is created (Figure 4.8, step 1) and published. This contract specifies which content c to deliver and some user preferences, such as the expected target quality (e.g., 1080p). Then the CPs are notified of the new CBC contract, and use it to create Content Licensing Contracts (CLCs) (step 2). The CLCs specify the price at which each CP is ready to sell content c to the EU, a reference to the CBC, and the maximum price for delivery. Next, once the CLCs are visible on the blockchain, TEs respond by publishing Content Distribution Contracts (CDCs) (step 3), which specify the cost they are willing to charge for delivering content c to the user and the reference to the CLC. Finally, the original CBC collects all the related CDCs and arbitrates toward the cheapest one (step 4). All other contracts are terminated, and the winning contract is used to implement the content delivery. Relevant technical information required to implement the contract, such as content ID, TE ID, and EU IP, are compiled in a Content Delivery Service Description (CDCS) document. We later detail how the contract is used to configure network service chains.

4.3.2.1.4 Content Delivery Session Monitoring

Smart contracts can implement a currency system to be used as a collateral means for ensuring the correct execution of the content delivery. Once the collaboration between

each actor is formalized in a CDC, the payers (EU and CP) transfer their due payments to the CDC, which behaves as an escrow account. Each partner sends a proof of activity to the delivery monitoring blockchain according to its role in the content delivery. For example, the CP could publish a cryptographic proof (e.g., as in the case of digital rights management) that entitles the TE to deliver the content. The EU publishes a proof of reachability of the content, whereas the TE publishes a proof of transmission. Once all the proofs are collected, the beneficiaries receive their payments. If the contract detects that a party has not fulfilled its duties, penalties can be applied, and some of the initial payment is refunded to payers.

4.3.2.1.5 Content provisioning

Content dissemination throughout the network is the key to reduce the price of the delivery. As each TE storing the content is able to propose a CDC contract, spreading a content widely increases the chance to find a short route between a TE and a given EU, who, in turn, lowers the delivery cost for the CP. Two options are available to increase the availability of a content: CPs can *push* them to the TEs or TE can *pull* the content according to the popularity.

Pulling: TEs can audit CBCs from the Content Brokering blockchain, to extract most popular content Id requested by EUs. TEs can thereby infer content popularity and decide which contents to store for increasing their chance of placing a winning CDC.

Pushing: CPs can offer an incentive to TEs to store specific contents through the Provisioning blockchain. The main idea, as proposed by Permacoin [Miller et al., 2014] or, more recently, by Retricoin [Sengupta et al., 2016], consists on using a proof of retrievability in the mining process, rewarding TEs that store a particular content. This technique can be used to disseminate contents even before they become popular, which speeds up their dissemination wrt to a passive pulling. The process of determining which content should be provisioned on which TE, can be addressed thanks to a resource prediction engine [Kryftis et al., 2016].

Our proposal suggests to foster competition by allowing several actors to provide their resources to the system. By decoupling the content delivery from the content licensing, we hence set up a much more diverse ecosystem, opening the possibility of including actual EUs (assuming the role of TE) in the participation of the content delivery process, by contributing with their spare resources. To conclude on the presentation of the blockchain aspects of our proposal, we discuss its possible governance models in the next section.

4.3.2.2 Discussion on Blockchain Governance Models

As the proposal relies on a fully decentralized agreement conclusion mechanism, we need a way to establish the respective liabilities of stakeholders in case of problems. As smart contracts are not legal contracts in essence, any litigation should be solved by proper prior legal agreements. Several models can be considered:

- **Chain of responsibility** Each actor contracts with a supplier, which is liable for the service it provides. CPs are liable toward EUs, TEs are liable toward CPs, and TEs are

liable toward ISPs. This solution is not very scalable as it implies having thousands of contracts.

- **Consortium** Actors create a consortium providing the legal foundations for the service [Buterin, 2015]. The consortium manages any liabilities centrally and automatically thanks to the blockchain. This model opposes the decentralization of transactions, but offers a more scalable alternative.

- **Decentralized autonomous organization** In this model [Jentzsch, 2016], legal aspects are directly managed on-chain by an organization the governance of which is defined by the code of smart contracts, bringing full decentralization and automation. However, the legal status of this type of business organization is still unclear.

Our proposal fosters competition by allowing several actors to offer their resources to the system and adjust their prices to match demand. By decoupling the content delivery from the content licensing, we set up a much more diverse ecosystem, by including actual EUs (assuming the role of TEs) in the content delivery process. However, constructing content sessions by using third party resources induces a challenge to current Internet architectures. In the next section, we describe how content sessions can be dynamically mapped to network service chains through network softwarization and the use of micro-services.

4.3.3 Instantiating the Model through Advanced Dynamic Network Service Chains

Once the brokering of content licensing and delivery is complete, the content session between the TE and the EU is implemented. Content sessions are on-demand, user-centric service chains deployed based on the specifications of the CDCS. The deployment of the service chain is shared between ISPs and TEs, the ISPs being responsible for steering the traffic of the EU to/from the TE domain, while the TEs implement both networking and service configuration of IP endpoints. TEs implement content delivery in several ways. We detail three complementary approaches: CDN, vCDN, and mCDN, detailed below. Figure 4.9 shows the deployment of these three types of TE. User 1 gets its content from a CDN, while User 2 uses a CDNaaS deployed in the ISP network. User 3 retrieves its content directly from user 4's mCDN.

We already covered CDN delivery in Section 3.2 and vCDN through CDNaaS in Chapter 3. We present an alternative third model, μCDN delivery, loosely derived from our previous work on virtual home gateways [Herbaut et al., 2015].

μCDN Delivery

Customer Premises Equipment (CPE) provides plenty of spare system and network resources that can be used for content delivery. With modern operating systems (GNU /Linux,

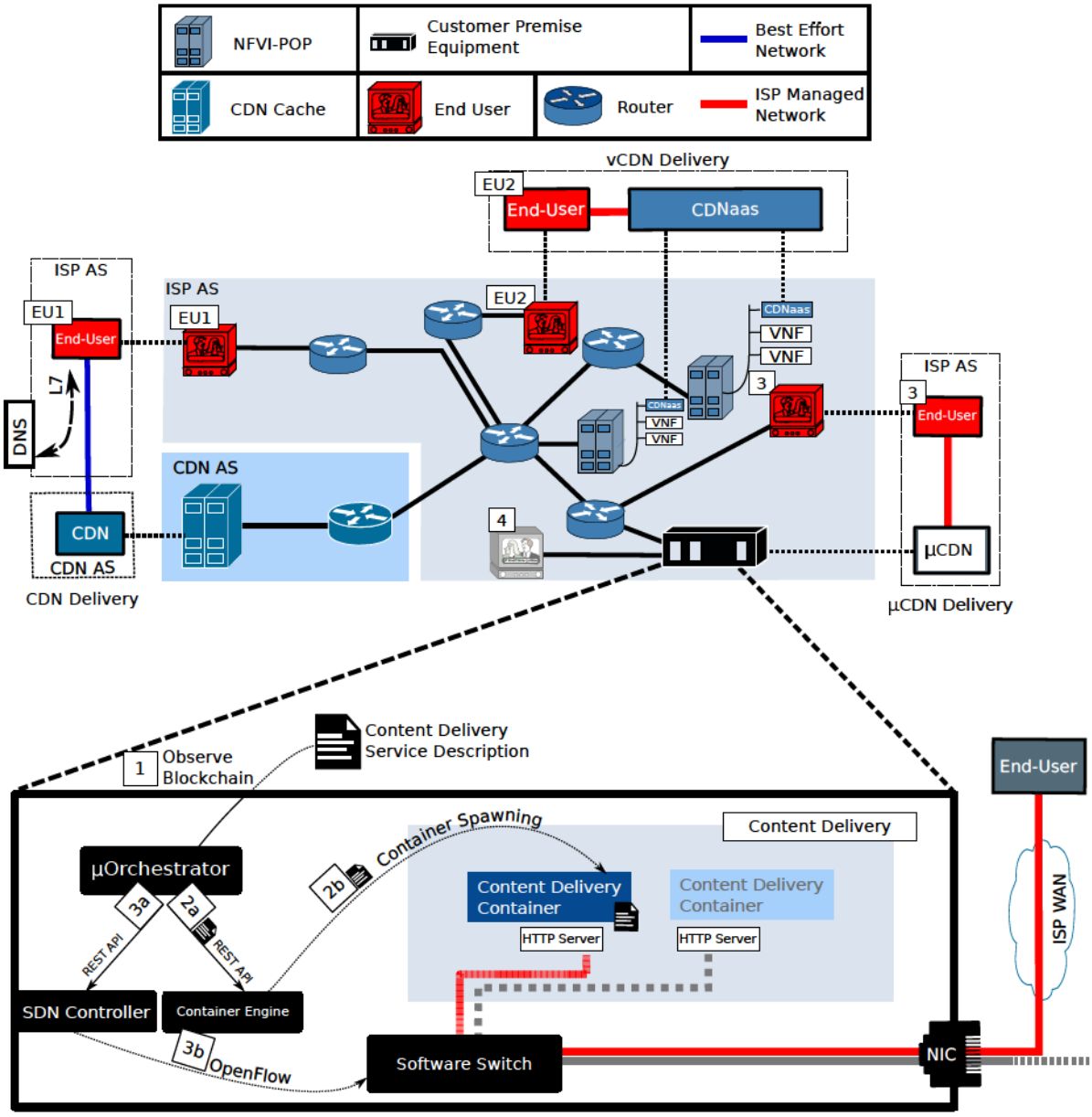


Figure 4.9: CDN, CDNaaS and μCDN services deployed in an ISP network

Android), they can support the deployment of new services and even Virtual Network Functions [David Minodier, and Gregory Dalle, Juniper, 2016]. Their small scale however requires downscaling the main concepts behind NFV.

Figure 4.9 shows the internal microservice [Fowler and Lewis, 2014] architecture used to implement the μCDN. The two key technologies we use to address the above-mentioned challenges are Linux Containers and SDN, as follows:

- Containers are lightweight virtualization mechanisms that bundle applications and their dependencies. With their reduced footprint and low CPU overhead, they are often considered in cloud edge architectures[Pahl and Lee, 2015]. In a very recent paper [Cziva and Pezaros, 2017], authors presented their concept of "Glasgow Network Functions", a container-based NFV platform exploiting lightweight container VNFs deployed as close to the users as possible. Their evaluation suggest that their solution can be applied to a variety of edge platforms, such as Internet of Things (IOT) devices or home gateways. network edge

- The SDN-capable software switch deployed on the CPE allows manipulating the containers connectivity in an OS-independent fashion thanks to the use of standard protocols such as OpenFlow.

Service deployment is triggered by the publication of the CDCS on the Blockchain (Figure 4.9 - (1)). It is retrieved by the μOrchestrator module, which spawns Content Delivery Containers running HTTP Servers able to stream the content (2a) and to configure their network (2b). CP may choose to use different technologies to license their content, from simple file to more complex Digital Rights Management (DRM)-based solutions. Adopting a microservice architecture, our solution keeps these implementation details in the Content Delivery Container and assures that, regardless of the underlying technology, the Content Delivery Service Description provides every resources needed (e.g., cryptography material). Finally, (3a) the μOrchestrator informs SDN Controller to update the network configuration that is, in turn, (3b) deployed by the Software Switch, so that the connection between the EU and the Content Delivery Container can be established.

4.3.4 Evaluation of the proposed model

4.3.4.1 Network Services Chain evaluation

We implemented a discrete event simulator with the SimPy Library[Matloff, 2008] to emulate content delivery sessions. We simulated 15,000 content session requests spanning over 25 minutes. For every requests, each TE that (1) stores the content and (2) has enough bandwidth to deliver the content asks for a delivery price assumed to be proportional to the number of hops between itself and the EU. The brokered price corresponds to the smallest price demanded by a TE. CDNs were assumed to host the entire content catalog whereas μCDN and vCDN pulled the content from CP by auditing Blockchain data and downloading the most popular contents. We used a real ISP topology of 2k nodes and 60k edges extracted from the Center for Applied Internet Data Analysis. Six CDNs were placed in a weighted random fashion at the most connected links, which correspond to the Internet Exchange Points on the operator topology. We then placed 500 Service Access Points nodes representing the user location in the network in a similar way, selecting the least connected links. Finally, 100 vCDNs and 500 μCDNs were randomly distributed among the nodes

with connectivity degrees in the middle range (40-90%) vCDNs capabilities were based on common virtual caching appliances specifications (1 TB of storage supporting 150 Mbps or 30 concurrent 720p streaming sessions) while μCDN capabilities were based on current customer premises equipment specifications (30 GB of Storage, 20 Mbps of upload speed or 4 concurrent 720p streaming sessions). Contents stored in μCDN and vCDN are purged according to a Least Recently Used rule. CDNs were assumed to support a large amount of concurrent connections (2.5 Gbps or 500 sessions). We assumed content popularity to follow a zipf distribution. The hop count is computed from the topology for vCDNs and μCDNs , however, for the CDN, we assume that 3 additional hops are used within the CDN network between the edge of the ISP network and the final server, corresponding to the average ISP graph distance.

Results of the experiments are presented on Figure 4.10 and 4.11. This first shows the respective shares of TEs. At the beginning, we see that every request is served by the CDNs, as they are still the only ones hosting the content. After 2 mins, once the popular contents are downloaded by the vCDNs, they also start delivering contents. The reason why vCDNs are privileged wrt CDNs is that they are spread more widely in the network, with a smaller average distance to EUs. After 3 mins, the μCDNs start serving content as well, and their share increase up to 12 mins, where they become the most used TEs. Again, this can be explained by a denser distribution of μCDNs in the network causing a lower hop count. After the 20 minutes mark, the shares stabilize. Despite their advantages, μCDNs only absorb half of the content requests. In fact, due to their limited capacity and storage, they are able to store and deliver only very popular contents. vCDNs store both very popular content and less popular contents and still account for a third of content session. Finally, CDNs absorb the long tail of contents, which are not popular enough to be stored by other TEs.

Another important benefit of our solution is the hop count reduction. Figure 4.11 compares the average number of hops between the selected TE and the EU when using all three TE types in conjunction, but also using only some of them. We can see from the figure that using only the CDNs yields to a higher hop count, stable over time. When complementing CDN with vCDNs, the hop count sharply decreases, as content gets stored near the edges of the network, and stabilizes near the 4 hops mark. When using both CDN and μCDN, the curve decreases slowly, as contents take more time to be provisioned in the edges. Finally, using all 3 TEs yields to the lowest hop count, with a fast drop at the beginning and a downward trend reaching the lowest value of our experiment.

4.3.4.2 Blockchain evaluation

4.3.4.2.1 Test environment

Our goal is to build a system where each content session is brokered on the Blockchain. For this reason, its *performance*, measured in terms of number of transactions processed per second, is the key to provide the content sessions quickly. At the same time, we envision the number of “clients” (EUs, CP and TEs) using the service to be high, so the Blockchain

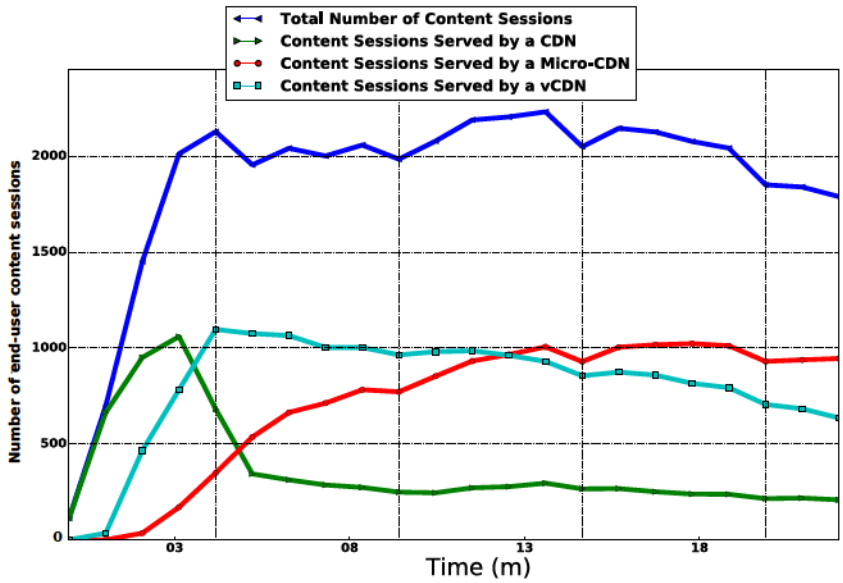


Figure 4.10: Respective TEs share for CDC

must ensure a good *node scalability*. Today, "permissionless" Blockchains based on Proof-of-work consensus offer great node scalability, but lack the required throughput (e.g., up to 7 tx/second with Bitcoin). On the other hand, Blockchains based on advanced Bysantine Fault-Tolerant (BFT) state-machine replication protocols offer excellent performance in terms of throughput and latency but require all nodes to know the IDs of all other nodes [Vukolić, 2015]. In our case, we used a “permissionned” Blockchain as the nodes processing the transactions do not need to be anonymous.

The need for performance and Smart Contracts support compelled us to use the open source project Hyperledger-Fabric[21]. This Linux foundation project can be used to build Blockchain solutions with a modular architecture to deliver flexibility and scalability. It provides pluggable consensus algorithms (by default PBFT) and simple Smart Contract implementation in Go or Java.

The critical aspects of the brokering mechanism is the time needed to converge toward the optimal Content Delivery Contract, involving the EU, the CP and the TE. This delay affects the EU QoE, as the content delivery session can start only after the optimal CDC is computed. A lot of contracts are published in the Blockchain, for example, if we assume that there are 10 CP and 100 TEs, then up to 10×100 contracts will be published.

Considering this, the evaluation is focused on the Content Brokering Blockchain as it is the most time-sensitive and subject to scalability issues.

We deployed the solution with Hyperledger-Fabric configured with the PBFT consensus, as shown on Figure 4.12. We then paired EU applications (publishing CBC), CP applications

[21] www.hyperledger.org/projects/fabric

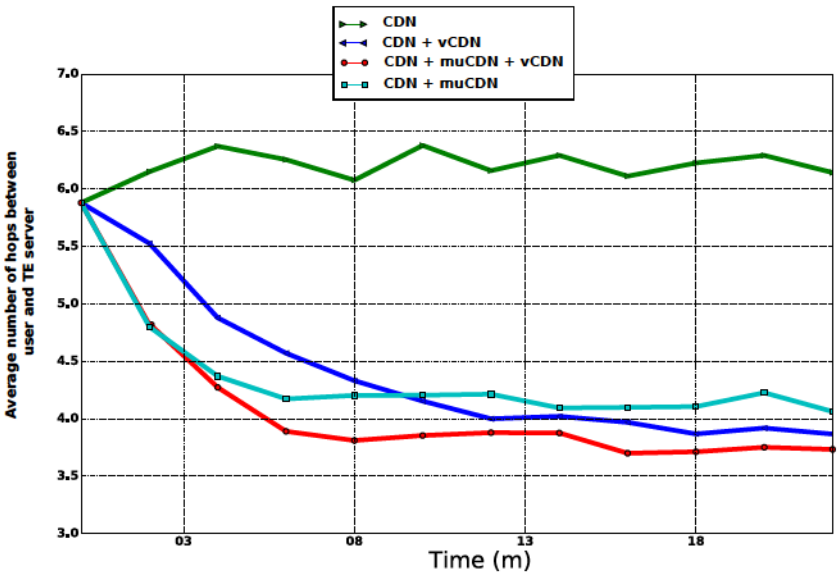


Figure 4.11: Average price for content delivery

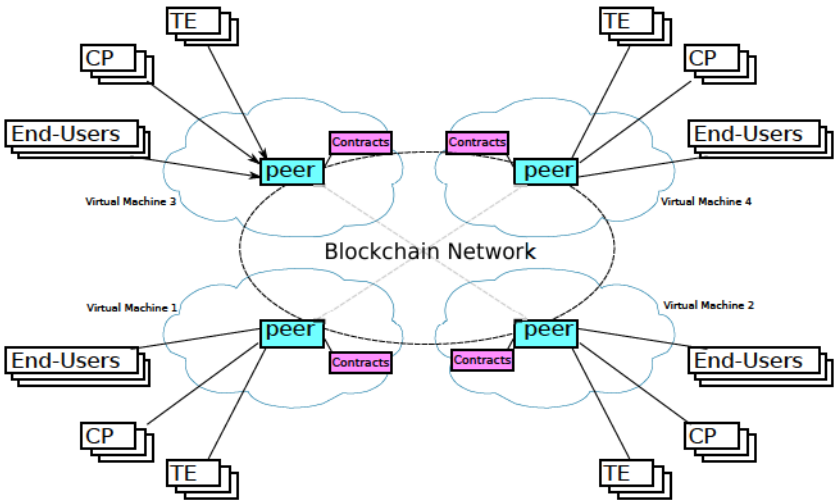


Figure 4.12: Testbed

(reading CBC from the Blockchain and responding by publishing CLCs) and TE applications (reading CLCs and publishing CDCs), the Blockchain validating peers, which are the nodes responsible for running the consensus, validating transactions, and maintaining the ledger.

Each user was configured to send 10 requests per minute. We then computed the average time needed to obtain the optimal CDC, or convergence time. We varied the number of TEs agents, the number of CP being fixed at 10.

The results presented in Figure 4.13 show that for 50 TEs, the convergence time is below

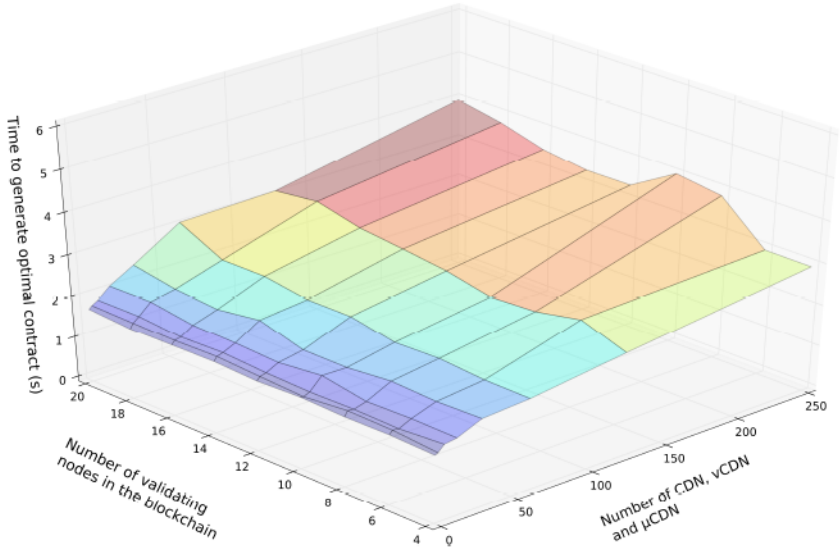


Figure 4.13: Performance and scalability experiment

2s. This time increases for higher values of TE, reaching 4 seconds in the worst-case scenario of 250 TEs, which remains acceptable.

4.3.4.2.2 Discussion on scalability

Performance studies of blockchain systems have so far been restricted to public blockchains. These approach consider the impact of block sizes and network propagation time on the overall throughputs [Croman et al., 2016] but cannot be applied in our case as we use a permissioned blockchain.

The Blockchain network is composed of validating nodes that run the Smart Contracts and append blocks to the chain once the consensus is reached. They are also used to query the state of the Blockchain by clients. Increasing the number of validating nodes has two antagonist effects: (1) each node serve less clients, reducing the average number of requests per node and (2) the quorum needed for the consensus is increased, impacting directly the number of messages shared in the network. On our testbed, the number of nodes increased slightly the convergence time. This is due to the rather good networking performances of our cloud instances, located in the same availability zone. In a production deployment, nodes would not be collocated to improve resiliency, and the performance may be even more impacted. In any case, significant performance improvements can be achieved by fine-tuning the operational parameters exposed by the blockchain provider (e.g. batch size, timeout). This require using specific benchmarking tools that can inject smart contracts and repeatable workload to the system under test, as proposed in [Dinh et al., 2017].

The new release of Hyperledger-Fabric 1.0 and recent research papers such as [Li et al., 2017] promote new architectures that support parallelizing the validation of trans-

action through their endorsement by only a subset of nodes. In this perspective, transactions are managed on sub-chains supporting fine-tuned consensus algorithms, so improving scalability.

4.4 Conclusion

In this chapter, we have proposed two collaboration schemes that can be used to improve the quality of content for users, reduce the costs and foster collaboration between ISPs, CPs and CDN providers.

The CDNaaS model provides means for the vCDN customers to define an SLA describing the expected quality of delivery of their content. By deploying their own CDN implementation within the ISP network and managing it through an overlay, they keep control of their core business while limiting the upfront costs of deploying their own dedicated delivery appliance. The model also benefits the ISPs by improving the content of OTT services to their customer, saving on peering costs and creating a new source or revenue, that can be used to fund further investments. By providing high-level APIs and sharing selected network control knobs to their vCDN customers, they keep their physical network private while allowing a collaboration that ultimately benefit the EUs.

The User-centric proposed model further improves the CDNaaS model by providing each content session with the best mix of CPs, content delivery technical enablers and network providers. Allowing a fine-grained selection of competing actors to provide the best experience is only possible if the system is transparent and fair. The Blockchain and smart contracts provide a robust and trustworthy mechanism to reach these goals.

Our work is a first step toward the necessary collaboration of the content value chain and show how the agility and openness of network softwarization can overcome the issues on content distribution in access networks. Needless to say, many challenges are not solved yet, such as the massive scaling required to process all the content delivery and the concrete CDNaaS deployment in the ISP Points of Presence. For these reasons, the following chapter investigates the problem of mapping the required CDNaaS resources expressed in the vCDN SLA with the resources of the physical ISP network.

CHAPTER

5 Deployment and Optimization of Virtual Content Delivery Networks

In the previous two chapters, we examined the implementation of CDN-as-a-Service (CDNaaS), a Virtual Content Delivery Network Service, and discussed the possible schemes making possible a collaboration between Content Providers/Content Delivery Network (CDN) operators and Internet Service Provider (ISP). However, the deployment of such service has practical implications that we did not cover: (1) how does the ISP and Virtual CDN (vCDN) customer agree on service specifications and pricing (2) how can the ISP minimize its deployment costs while offering the required quality of service and (3) how the vCDN customer predicts its service usage and minimizes its service costs. In this chapter, we present more formally the *vCDN Service Level Agreement (SLA)* and show how it can be used by the vCDN customer to specify the service and by the ISP to compute its price before deploying it on its network.

Our contributions are three-fold: First, we specify the vCDN SLA and explain how it can be seen as a precursor of a Service Graph for a Service Chain Embedding problem in a Virtual Network Function (VNF)-as-a-Service approach; Second, we present an adaptive algorithm that solves the vCDN Embedding Problem derived from the SLA; Third, we generalize this approach when SLAs are dynamic.

In addition, several online assets have been produced (and are available in open-source) in order to conduct our research:

- An interactive tool highlighting the embedding algorithm which runs the actual code produced: http://demo-girafe.nextnet.top/
- A dataset collected to study the multi-site dynamic SLAs: http://data.nextnet.top/

5.1 Introduction

The SLA expresses a set of requirements and business agreements between the vCDN customer and the ISP, Is is a first important step in describing the service that the ISP needs to implement. As mentioned in Section 4.2.2.2, the vCDN customers do not have a complete vision of the underlying physical network where their service will be deployed. For this reason, the exact number of VNFs (Virtual Media Gateway (vMG) and VStreamer in the case of CDNaaS), their location and the physical paths supporting the data flows between each each one is not specified in the SLA. In fact, it is in the best interest of the ISP to select the Point-of-presences (POPs) and routes that will be used to support the vCDN Service, in order to minimize the cost of the service and increase the number of embedded SLAs and its profit. In other words, the vCDN customer is responsible for describing the global behavior or intent of the vCDN through the SLA, while the ISP is in charge of defining the concrete service layout.

The translation of a high level SLA to a service graph definition is only the initial step of the wider problem of Resource Allocation (RA) in Network Function Virtualization (NFV). In [Herrera and Botero, 2016], Herrera further breaks down the RA problem in several sub-problems, that we all cover in this chapter.

- VNF chain composition, which is the result of the processing of service description presented before;
- VNF forwarding graph embedding, which deals with the allocation of virtual resources to nodes (mapped to substrate nodes) and links (mapped to substrate paths) where the nodes are crossed in a specific order by the flow;
- VNF scheduling, where embedding requests and mappings are organized to improve the deployments of the services and to reduce the allocated network and system resources.

In this section, we present the background on these three important problems.

5.1.1 Related work on VNF Chain Composition

The challenges of describing a service potentially containing VNF is surveyed by Garay et al. in [Garay et al., 2016]. The authors propose a straw-man model to describe both the physical and service networks as graphs, with a definition that allows them to be processed by several hierarchical orchestrators.

This formalism, however, does not necessarily cover the generation of the service model. In [Cohen et al., 2013], authors propose to create network blueprints for deterministic and verifiable specification of network functionality. Expanding this approach to VNF chain, Callegati et al [Callegati et al., 2017] propose a way to specify service chains as a vendor-independent intent submitted to a northbound Application Programming Interface (API)

responsible for translating the service in a multi-domain Software Defined Networking (SDN) deployment. The principles of such a northbound API have been clarified by the Open Network Foundation [Foundation, 2016]. A more semantic approach was proposed in [Mehraghdam et al., 2014a] where authors proposed a context-free grammar to formalize chain requests.

Our work goes beyond the cited work by also taking into account the financial aspects. As we saw in 2.3.4, the T-NOVA marketplace uses a brokering mechanism to select the most cost-effective VNF implementation according to the SLA. We also consider the algorithmic impacts of the service chains generation, keeping in mind that the computation of the optimal chain embedding can be slow, especially for huge operator networks with several VNF providers offering iso-functional implementations.

5.1.2 Related work on VNF forwarding graph embedding

Most of the existing work maps NF-graphs onto a single substrate network, which is usually assumed to be a data-center network (*e.g.*, with a fat-tree topology). In this respect, authors in [Sahhaf et al., 2015] investigate gains in terms of NF-graph mapping by breaking down NFs into a set of elementary NFs, in order to increase the search space for feasible solutions. Cohen et al. [Cohen et al., 2015] formulate NF placement as a facility location and generalized assignment problem (GAP), and propose approximation algorithms aiming at latency and NF setup cost minimization. Mehraghdam et al. [Mehraghdam et al., 2014b] present a solution to the NF-graph mapping problem, while considering different optimization goals. The authors further define a model for NF-graph transformations (*i.e.*,, NF reordering, replication, or merging) to optimize the NF placement for the provider.

Authors in [Lukovszki et al., 2016] tackle a variant of the NF mapping problem, *i.e.*, the placement of NFs on a network, while ensuring that each path between a pair of end points has at most one NF assigned. The proposed approximation algorithm further facilitates the incremental deployment of NFs. However, this work does not account for service chaining. In [Lukovszki and Schmid, 2014], the authors study the online variant of the service mapping problem. In this respect, they propose an exact method that maximizes the request acceptance rate, while fulfilling constraints in terms of path length for the service chain. Bari et al. [Bari et al., 2015] derive an Integer Linear Programming (ILP) formulation and heuristic algorithm for service mapping with the objective of operational cost and resource fragmentation minimization. STRATOS [Gember et al., 2013] and CloudNaaS [Benson et al., 2011] also propose heuristic mapping algorithms that seek to minimize inter-rack traffic within datacenter networks.

Recent work has focused on service mapping across multiple providers. Nestor [Dietrich et al., 2015] decomposes multi-provider service mapping into NF-graph partitioning among providers and NF-subgraph mapping onto datacenter networks. Nestor further uses ILP formulations for the solution of the NF-graph partitioning and the NF-subgraph mapping problems. While Nestor provides exact solutions to both problems,

the use of ILPs leads to increased time complexity and poor scalability. MIDAutonomous System (AS) [Abujoda and Papadimitriou, 2015] also addresses service mapping across multiple providers; however, MIDAS is primarily focused on network processing setup along the traffic path, assuming a wide-scale deployment of middleboxes in the network. In this respect, MIDAS proposes a signaling protocol for the discovery of middleboxes deployed in the data path. In terms of NF placement, MIDAS couples Secure Multi-Party Computation (MPC) with a heuristic algorithm for order-preserving NF assignment across multiple providers.

In this thesis, we developed an embedding algorithm that takes into account the specifics or the vCDN-as-a-Service embedding problem:

- VNFs cannot be reused across tenants, as each customer can implement the vCDN in a specific way;
- The vCDN service chain has several ingress points and several optional egress points. Indeed, if the vCDN customer owns its own network, the service can span to all or part of the peering points where it is present;
- Path and node splitting must be supported at the Service Graph creation level;
- The scheduling of the VMs inside each POP is considered out of scope.

5.1.3 Related work on VNF Chain Scheduling

Considering that the NFV concept is still emerging, the VNF scheduling problem is an emerging topic. In [Riera et al., 2014] authors documented the concept and challenges, providing a formalization and an ILP formulation to minimize the execution time of Network Service. In the literature, this problem has been addressed from different perspectives: from energy efficiency [Bolla et al., 2014] to end-to-end service execution time [Mijumbi et al., 2015], revenue [Naudts et al., 2017] and request acceptance ratio [Huang et al., 2017]. The strategies to support the scheduling are also numerous, ranging from the simple node mapping [Mijumbi et al., 2015], to selecting a cloud provider in a multi-cloud environment [Bhamare et al., 2017], or providing the compute capability for short-life network service through a pool of Docker containers [Zhang et al., 2016].

In our case, the objectives differ since the implementation of the vCDN is different for each tenant. The stateful nature of CDNaaS also has practical implications: since the vStreamer consists in large cache storages, allocating a new VNF and provisioning the content is very time-consuming, hence a certain inertia in scheduling that should be taken into account through the inclusion of VM migration costs. Finally, minimizing the end to end delay is not necessary as long as the SLA delay is fulfilled.

The rest of the chapter is organized as follow: Section 5.2 presents two algorithms for service chain composition. Section 5.3 formalizes the Embedding problem, an ILP and

proposes an Heuristic. Finally, Section 5.4 covers the case where the SLAs are dynamic along with the implication on the scheduling and pricing of the service.

5.2 CDNaaS chain composition

As presented in Section 4.2, vCDN customer service requirements are expressed using a Service Level Agreement. The service created to fulfill this SLA needs to be embedded into the ISP's infrastructure. In this section, we essentially detail what is contained in an SLA, provide algorithms to express this SLA as Service Function Chaining (SFC) and present a linear programming formulation for the SFC embedding problem. Once the service is embedded, we show how it can be managed online thanks to the Element Manager API through the *vCDN :emAPI* introduced in Section 3.5.

5.2.1 Formalization of the SLA

Negotiating SLA with ISP brings significant advantages over traditional Cloud Providers' services, as the ISP clients control the network end-to-end. Usually, all-purpose Cloud Providers offer pre-configured Virtual Machine instance types specifying a certain number of Central Processing Unit (CPU), RAM and Storage. For example, the Amazon EC2 instance type *m4.xlarge* proposes 4 vCPUs, 16 GB of RAM , "High" Networking performances and "EBS " Storage volumes of various capacity and speed[1].

In our case, the SLA is designed to support the specific requirements of content delivery, and can be written using high-level concepts instead. Table 5.1 shows the SLA specification used to describe the vCDN service from the point of view of the vCDN customer.

First, the vCDN customer specifies the location of the End-Users (EUs) for which it wishes to get connectivity, the expected number of concurrent EUs N for the service and the target bitrate for video content (delivered over HTTP). These three inputs will be used by the ISP to compute the required TCP throughput to support the target bitrate B and the total bandwidth that needs to be allocated between the cache server and the users $T_b = B * N$. From the TContent Provider (CP) throughput, the ISP can then derive the maximal Round-trip time between a user and the cache server using the Mathis formula [Mathis et al., 1997] $\text{RTT} < \frac{\text{MSS}}{\sqrt{p}}$ with MSS the maximum segment size and p the probability of packet loss. The vCDN customer can optionally specify where its own server can be reached through peering points, so that part of the traffic toward the vCDN can be offloaded to them in best-effort mode (performance for this traffic is not covered by SLA though, as no maximal delay is enforceable for inter-AS traffic).

Second, the **VNF System Requirements** for the vCDN service, where the vCDN customer specifies how many VMs of a specific flavor are needed to implement each components (see

[1]https://aws.amazon.com/ec2/instance-types/

Table 5.1: CDNaaS SLA description

Metric Name	Description	Unit	example
Target Bitrate	Video bitrate experience by the EU	Mbps	2Mbps
Target Concurrent Users	How many users should be supported at the same time	#	15000
Target Client Groups (CG)	What groups of ISP EUs are targeted by the SLA	IP prefix, airport code[2]	109.15.0.0/16, cdg bod
CDN exit points (optional)	Peering points where existing legacy CDN server can be reached	peering points identifiers	IP address or IP prefix
VNF System Requirements	System requirements used for each type of VNF used in the service	Number of Virtual Network Function Component (VNFC) per VNFs	2 "m1.large" VMs for each 1,000 users for vStreamer
SLA duration	Beginning and end date of an SLA	Timeframe	2017-09-01T05:22:20Z 2017-10-01T08:00:00Z

Section 3.3.4 for an example of the impact of increasing the number of VMs on a VNF and Section 4.2.2.1 for an example of such a configuration through the marketplace). In that case, the ISP can plan the provisioning on dedicated or shared instances [Rimal et al., 2009]. Dedicated instances provide a complete isolation of the vCDN Customer VNFs and predictable performances that are not influenced by other tenants usage of the underlying hardware. Shared instances are easier to manage and to optimize since resources are shared between tenants. However, the ISP needs to design a policy aiming at honoring the performances required by the SLA, which can be difficult to achieve [Yfoulis and Gounaris, 2009]. ISP resources planning is considered out of scope for this thesis, and we assume that the ISP always allocates dedicated virtual resources.

Finally, the SLA is valid for only a defined period of time, ranging from minutes to months depending on the business of the vCDN customer.

5.2.2 From SLA to Service Function Chain

While the SLA provides the means for CDN operators to formulate their business needs, it does not specify technically how ISPs should perform the deployment into their network.

[2] Amazon Web Services names their Edge Locations after the closest International Airport IATA Code.

The following paragraphs discuss the translation of the SLA into a Content Delivery SFC request.

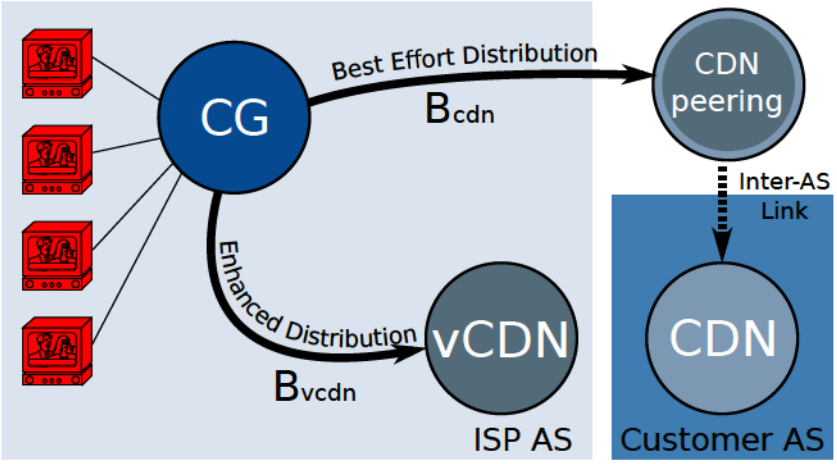


Figure 5.1: Canonical Model

5.2.2.1 The Canonical Service Model

Once the vCDN Customer has expressed the SLA, the underlying vCDN service can be represented under its canonical form. Starting from Client Groups (CG) (representing the geographical area towards EUs), best effort connectivity B_{CDN} towards vCDN customer own CDN network and enhanced connectivity B_{vCDN} towards vStreamer servers are depicted in Figure 5.1.

On the one hand, the $CG \longleftrightarrow CDN$ flow represents the traditional content distribution path: packets flow through the ISP AS and reach the separate AS at a peering point. On the other hand, the $CG \longleftrightarrow vCDN$ flow is targeted at a Network Function Virtualization Infrastructure Point of Presence (NFVI-POP) within the ISP AS. It represents the enhanced distribution path on which the SLA is applied. It has tighter network constraints (low delay, high bandwidth) and is aimed to deliver higher quality videos.

This canonical model effectively represents the service as instructed in the SLA, but does not necessarily take into account the implementation details of the service, nor the concrete deployment on the ISP network. Indeed, as soon as the SLA is agreed by both the ISP and the vCDN customer, the former has every latitude to implement it in a way that optimizes its business targets (e.g., cost minimization, profit maximization, fair balance between customers). The following section shows how a SLA can be concretely implemented by the Service Function Chain, based on our analysis presented in chapters 3 and 4, and introduces the formalization of the vCDN Service Chain.

5.2.2.2 Concrete implementation of the Canonical Service

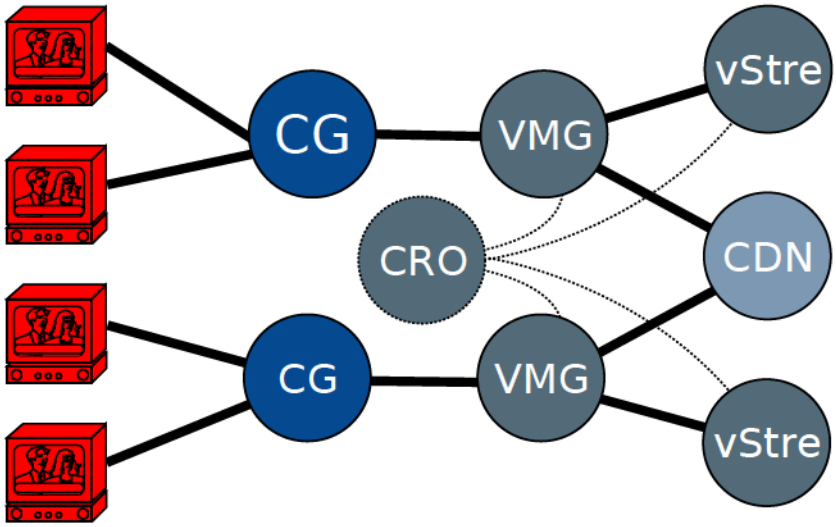


Figure 5.2: Concrete Implementation of the CDNaaS Service using SFC

As we saw in section 4.2.1, the vCDN service can be implemented using VStreamer VNF for content delivery and the Caching and Routing Orchestrator to manage the service. Further breaking down the content delivery part, Sections 3.3.2 and 3.3.3 showed that our solution uses another component: the Virtual Media Gateway, to handle efficiently the request routing at the HTTP level by performing Deep Packet Inspections at the Universal Resource Locator (URL) and HTTP header levels. The VMG is of particular interest considering the fact that only the traffic on which the vCDN customer wishes to apply the enhanced distribution should be targeted at the VStreamer.

With the addition of the VMG, the canonical model can now be reformulated as an SFC. Figure 5.2 shows a service graph derived from a SLA for which each CG can be connected to one or several vMGs, which route the request and response from one or several vStreamers and CDNs, depending on the configuration deployed by the Caching and Routing Orchestrator (CRO) VNF presented in Section 3.3. It should be noted that both vMG and Virtual Streamer (vStreamer) are "data plane" VNFs which process the traffic to and from the EU and are subject to the SLA constraints in terms of bandwidth and latency, while the CRO is a "control plane" VNF which receives and sends management information to and from the other VNFs, without necessarily having such constraints. For simplicity, we only consider data plane VNFs on the Service Function Chain, the problem of placement of the CRO can be related to the one of SDN Controller [Heller et al., 2012].

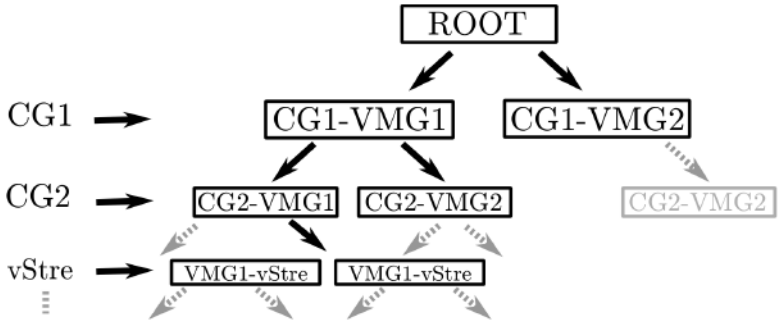


Figure 5.3: Building a tree with Service Edges.

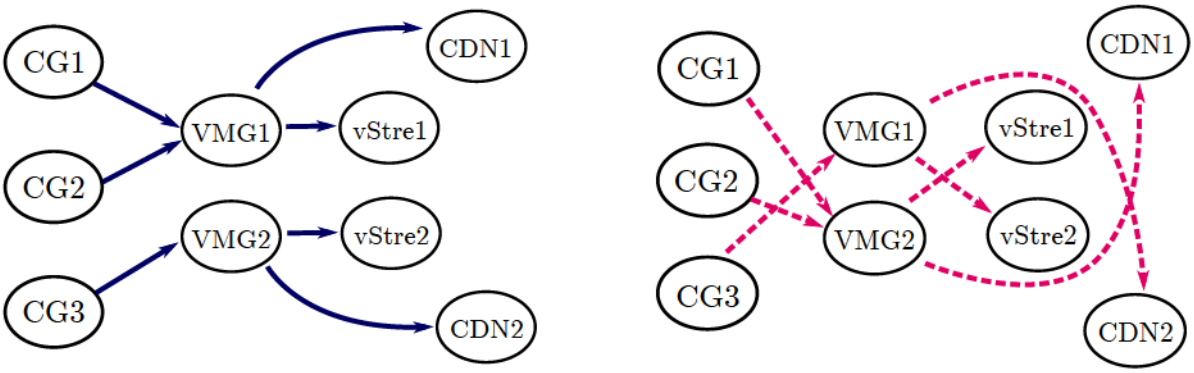


Figure 5.4: 2 Service-isomorphic graphs.

5.2.3 Service Graphs Generation Algorithms

In the previous section, we let the cost function use edge bandwidth $b_{i,j}$ and VNF cost c_i as decision variables, which make the objective function non-linear. A possible way to solve this problem is to pre-compute the bandwidth on the possible Service Graphs associated with the canonical model, and choose the service with the lowest cost. We consider that each Client Group is connected to exactly 1 VMG, and that each VMG is connected exactly to 1 vStreamer and 1 CDN.

5.2.3.1 Exhaustive Service Graph Generation

To build each possible Service Graph, we first create the service nodes (vMG, vStreamer, CDN peering point) and enumerate all the possible service edges combinations by constructing a tree whose nodes are all the possible edges of the service graph (Figure 5.3). Each layer of the tree represents outgoing edges for a service node. Once the tree is generated, we backtrack the service graphs from each leaf.

Since this process generates a very high number of possible services, we use the following method to reduce it only to meaningful ones.

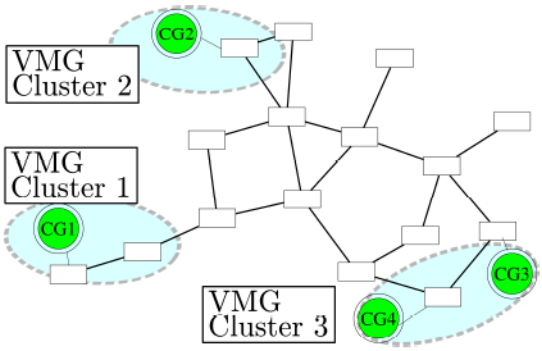


Figure 5.5: Phase I: Assigning vMG to CG

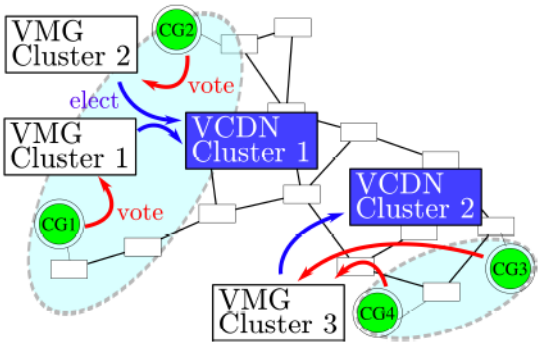


Figure 5.6: Phase II: Assigning vStreamer to vMG

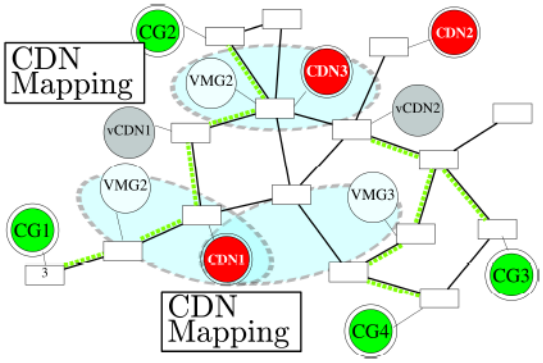


Figure 5.7: Phase III : Partial Embedding without CDN, Assigning CDN to VMG

When considering a service graph G^S, two types of service nodes exist: (1) Mapped nodes (Client Groups and Legacy CDN peering points) for which the mapping on the physical topology graph is *already known* as it is specified in the SLA, and (2) Un-mapped Nodes (vMG and vStreamer) for which the mapping on the physical topology is computed during

the optimization phase. In the backtracking phase, we must generate only meaningful services: for instance, the two graphs presented in Figure 5.4 are equivalent when renaming un-mapped nodes, i.e., vMG1 to vMG2 and vStre1 to vStre2 and vice-versa. A more formal definition is given thereafter.

Definition 1 *Two service graphs are said Service-Isomorphic if there exists an edge-preserving bijection between un-mapped nodes (i.e., VNFs) of the same type (i.e., vMG or vStre).*

It is obvious that the mappings on the physical topology obtained from two Service-Isomorphic Graphs are equivalent, by applying the same renaming scheme on the mappings and on the service graphs.

We now define the full set of service to be considered during the optimization phase:

Definition 2 *For a given SLA S, a Full Service Graph Class $\mathscr{C}_S(n,m)$ is the set of all the connected service graphs having up to n vMGs and up to m vStreamers, which are not pairwise Service-Isomorphic.*

Once the Full Service Graph Class is computed, we compute $b_{i,j}$ and c_i for each service graph and use them as parameters of the cost function (5.7), which becomes linear with respect to service graphs. The embedding problem is eventually solved by computing the optimal mapping for each service graph, and by selecting the cheapest among them.

5.2.3.2 Heuristic for Service Graph Generation

Due to the necessary optimization of a large number of service graphs in $\mathscr{C}_S(n,m)$, optimization becomes quickly intractable for 5 Client Groups or more. To circumvent this, we generate a much smaller Reduced Service Graph Class $\hat{\mathscr{C}}_S(n,m)$ with a service topology generation heuristic that uses the characteristics of the physical topology to decide how to generate service edges efficiently. In the following, we present a three-step service graph generation procedure.

Some service nodes are already mapped in the SLA, like CG and CDN peering points. We use this information in Phase I to generate vMG Clusters that, for a given number of vMGs, minimize the intra-cluster delay between CGs, as depicted in Figure 5.5. We repeat the procedure in Phase II (Figure 5.6), by creating vStreamer Clusters that, for a given number of vStreamer, minimize the intra-cluster delay between CGs. Note that the number of vStreamer clusters and vMG clusters is not the same in the general case.

We now have a Partial Service Graph $\tilde{G^S}$ with CG, vMG and vStreamer nodes. In Phase III, we compute $\tilde{G^S}$ delays and bandwidths, neglecting the CDN edges. We use the ILP to generate a partial mapping $\tilde{\mathscr{M}}$, which provides estimates for vMG and vStreamer nodes mapping. With those estimates, we are able to determine, for each vMG node, which CDN peering point is the closest, as shown in Figure 5.7. Based on this information, we finally create the service edges between vMGs and CDNs and compute $\hat{G^S}$. We save $\tilde{\mathscr{M}}$, as it will be used by the re-optimization feature of our ILP toolkit to compute the full mapping $\mathscr{M}$ faster.

5.3 CDNaaS Service Graph Embedding

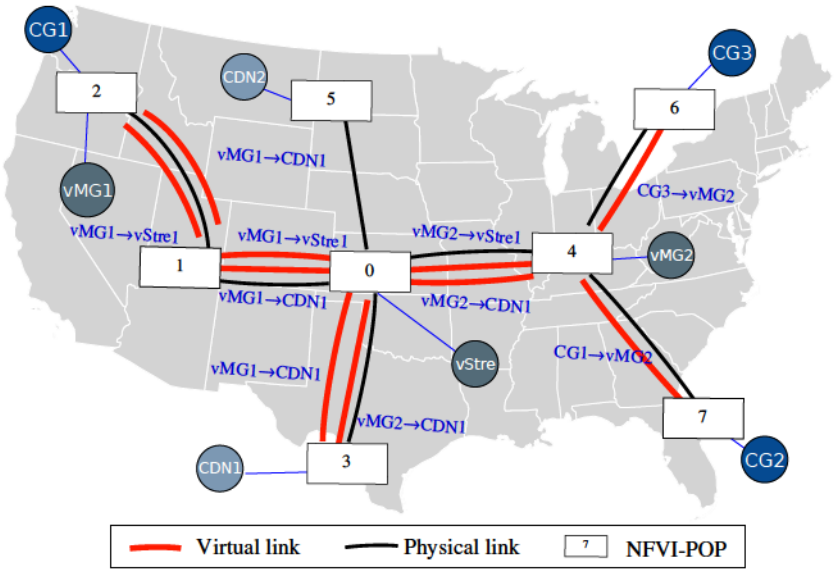


Figure 5.8: Mapped VCDN Service

Once the service graph is computed from the SLA, the ISP can embed it into its network. This problem is known as SFC Embedding [Quinn and Nadeau, 2015] and consists in mapping a service graph $G^S = (N^S, E^S)$ to the ISP physical network graph $G = (N, E)$. Figure 5.8 shows such a mapping, where service nodes are mapped onto physical nodes and service edges are mapped onto physical edges.

In addition to fulfilling the constraints expressed in the SLA, the ISP also aims at achieving its business goals. In our work, we consider that the ISP aims at maximizing its profit by minimizing its embedding costs. To this end we introduce in the next section, the cost function, the generation of service graphs and the linear model assuring that the SLA constraints are fulfilled within the embedding process. As each vCDN customer deploys the vCDN service using their own VNFs implementation, no VNF is shared between tenants and each SLA deployment results in the embedding of a new SFC. We formulate SFC embedding as an optimization problem, leveraging on existing work from the Virtual Network Embedding (VNE) literature [Chowdhury et al., 2012], [Fischer et al., 2013].

5.3.1 ILP Embedding Algorithm

This section highlights the two components of the ILP formulation: the constraints in Section 5.3.1.1 and the cost function in Section 5.3.1.2.

5.3.1.1 Formulation of SLA Constraints

Table 5.2: Notations

Symbol	Domain	Description
		Domains and sets
$\mathscr{P}^{(i,j)}_{\mathscr{M}}$	$N^S \times N^S \mapsto (E)^n$	set of the service edges linking i to j for $\mathscr{M}$
$\mathscr{D}^{i,j}_{\mathscr{M}}$	$N^S \times N^S \mapsto \mathbb{N}$	delay between service nodes i and j for $\mathscr{M}$
N^S_{GC}	N^S	set of CG nodes.
N^S_{vCDN}	N^S	set of vStreamer nodes.
		Decision variables
$y^{i,j}_{u,v}$	$E \times E^S \mapsto \{0,1\}$	$\begin{cases} 1, & \text{if } (i,j) \text{ is mapped } (u,v) \\ 0, & \text{otherwise} \end{cases}$
x^i_u	$N \times N^S \mapsto \{0,1\}$	$\begin{cases} 1, & \text{if } u \text{ hosts service node } i \text{ for } \mathscr{M} \\ 0, & \text{otherwise} \end{cases}$
		Physical Topology and Service parameters
$d_{u,v}$	$E \times E \mapsto \mathbb{R}^+$	the delay for edge (u,v) of node
d_i	$N^S \mapsto \mathbb{R}^+$	the delay for edge (u,v) of node
$b^S_{i,j}$	$E^S \times E^S \mapsto \mathbb{R}^+$	required bandwidth between i and j.
$b_{u,v}$	$E \times E \mapsto \mathbb{R}^+$	available bandwidth between u and v
c^S_i	$N^S \mapsto \mathbb{R}^+$	required computing resources for (i).
c_u	$N \mapsto \mathbb{R}^+$	available computing resources on NFVI-POP u.
$\delta(u)$	$N \mapsto N$	neighbors of u on outgoing links

We formulate the optimization problem using common network-flow notations, summarized in Table 5.2. We use two main binary decision variables, x^i_u to denote that the service node i is deployed on NFVI-POP u and $y^{i,j}_{u,v}$ to denote that physical link (u,v) is used to support all or part of the service edge (i,j).

First, we must ensure (5.1) that each service node is mapped to an NFV I-POP.

$$\sum_{u \in N} x^i_u = 1, \forall i \in N^S \tag{5.1}$$

Contrary to traditional hardware devices, VNFs are elastic and can adapt to varying workloads by automatically extending their footprint through automatic scale-out. ASthe design of our vCDN permit (see Discussion in Section 3.4), augmenting the capacity of a VNF only requires increasing the number of its VNFC through the orchestrator/VNF Manager. As each

VNFC is implemented as a Virtual Machine (VM), the system capacity needed by a VNF i can be expressed as a number of VMs of a certain flavor (see section 4.2.2.1). We assume that each VNF is dimensioned from the start to the maximum capacity needed to fulfill the SLA, thus requiring c_i^S VMs from the NFVI-POP where it is mapped. Eq. (5.2) makes sure that the NFVI-POP u can host the number of VMs required by each VNF i it hosts.

$$\sum_{i \in N^S} x_u^i \times c_i^S \leq c_u, \forall u \in N \tag{5.2}$$

Each service edge is possibly supported by several physical links, so by applying (5.3), we make sure that bandwidth $b_{u,v}$ can satisfy the demand for each mapped service edge $b_{i,j}$.

$$\sum_{(i,j) \in E^S} y_{u,v}^{i,j} \times b_{i,j}^S \leq b_{u,v}, \forall (u,v) \in E \tag{5.3}$$

Delay constraints (5.4) assure that $\forall s \in N_{\text{GC}}^{\text{S}}$, and $\forall t \in N_{\text{vCDN}}^{\text{S}}$, we do respect the maximal end-to-end delay $\mathscr{D}_{\mathscr{M}}(s,t)$ on each path $\mathscr{P}_{\mathscr{M}}^{(s,t)}$ joining the CGs to the vSreamer. For this, we take both physical links transmission delay $d_{u,v}$ and service node processing delay d_i into account.

$$\sum_{(i,j) \in \mathscr{P}_{\mathscr{M}}^{(s,t)}} \sum_{(u,v) \in E} y_{u,v}^{i,j}.d_{u,v} + x_u^i.d_i \leq \mathscr{D}_{\mathscr{M}}(s,t) \tag{5.4}$$

We also make sure that the service does not loop on the physical topology (5.5) by assuring that no more than one outbound link (u,v) is present for the same service edge (i,j).

$$\sum_{v \in \delta(u)} y_{u,v}^{i,j} \leq 1, \forall (i,j) \in E^S, \forall u \in N \tag{5.5}$$

Finally, we apply the flow conservation constraint (5.6).

$$\sum_{v \in N} y_{u,v}^{i,j} - y_{v,u}^{i,j} = x_u^i - x_u^j, \forall (i,j) \in E^S, \forall u \in N \tag{5.6}$$

We try to tackle nodes and edges assignments at the same time, so the problem is known to be $\mathcal{NP}$-hard [Amaldi et al., 2016]. In the next subsection, we detail the cost function and the generation of the service graphs according to it.

5.3.1.2 Formulation of the cost function

We aim at minimizing the total cost of mapping $\mathscr{M}_S$, subject to constraints (5.1)-(5.6) derived from the SLA S. Each ISP has a different cost structure, however, here, we consider the 2 main costs: network $p_{net}(\mathscr{M}_S)$ and hosting costs for the VNF $p_{VNF}(\mathscr{M}_S)$. We define the cost function as follows:

$$p(\mathscr{M}_S) = p_{net}(\mathscr{M}_S) + p_{VNF}(\mathscr{M}_S) \tag{5.7}$$

We assume that the network cost is proportional to the consumed bandwidth on all enhanced service paths (CG-vStre):

$$p_{net}(\mathscr{M}_S) = \sum_{(u,v)\in E} \Big(\sum_{(i,j)\in E^S, j\notin CDN} y^{i,j}_{u,v} * b_{i,j} \Big) \tag{5.8}$$

We do not take into account best effort paths (vMG-CDN) in our model for cost computation, since the ISP would support them even without the vCDN service.

Costs for VNF i are assumed to be linear functions $c_i(.)$ of the bandwidth transiting through or targeted at them.

$$p_{VNF}(\mathscr{M}_S) = \sum_{j\in N^S} c_i\Big(\sum_{i\in\delta^+(j)} b_{i,j} \Big) \tag{5.9}$$

The next section present two alternative embedding algorithms based on heuristics.

5.3.2 Heuristic Embedding Algorithms

The ILP formulation has the advantage of being easy to understand and implement, thanks to the availability of high performance open source solvers. However, it lacks scalability as the time to convergence is exponential with the size of the network and the size of the service graph, which can be an issue for operator networks. For this reason, we developed 2 alternative embedding algorithms that can produce sub-optimal solution for wider networks: the Dummy algorithm and a genetic algorithm.

5.3.2.1 Dummy Embedding Algorithm

We start by describing a simplistic algorithm that can produce constrained embedding, without taking into account the cost function.

This algorithm simply takes every service node and maps it to a random physical node. It then creates the edges from the mapped parent of the node by computing the shortest path using the Dijkstra algorithm [Dijkstra, 1959] on the physical graph, weighted with delays. It proceeds until all the nodes are mapped and finally returns.

Even if this algorithm does not take into account the cost function of the ISP, it can produce valid constrained mapping quite rapidly in $\mathcal{O}(V_N^2 \times V_{N^S}^2)$ (where V_N is the number of physical nodes and V_{N^S}=the number of service nodes).

Another advantage for this formulation is that a mapping $\mathscr{M}$ can be described only by its Node mapping (that is by knowing on which physical node each service node is deployed). It means that a dummy mapping can be written as $\mathscr{M}_{\text{dummy}} = \mathscr{M}(n^s \rightarrow n, n^s \in N^S, n \in N)$, the dummy algorithm being in charge of computing the Edge mapping (the physical links are used to support a service link). Using this formulation, in the next section, we derive a genetic algorithm implementation of the initial optimization problem.

Algorithm 1: Dummy Embedding Algorithm

Data: Service Graph $G^S = (N^S, E^S)$, Physical Network Graph $G = (N, E)$
Result: A Mapping $\mathscr{M}$

1 update $\mathscr{M}$ with pre-mapped nodes (GC and CDN);
2 **for** $\hat{n}^S \in N^S$ **do**
3 $N_{\text{avail}} \longleftarrow$ Get All Nodes with Enough VMs for $\hat{n}^S$;
4 $P_{\hat{n}^S} \longleftarrow$ Get All parents of node $\hat{n}^S$;
5 **while** $\hat{n}^S$ *is not mapped* **do**
6 choose n_{random} randomly in N_{avail} (without replacement);
7 **for** *each* $n_p^s \in N_{\hat{n}^S}$ **do**
8 $G_{n_p^s,\hat{n}^S} = (N, E_{n_p^s,\hat{n}^S})$ such that: $E_{n_p^s,\hat{n}^S} = \{e \in E, \text{bw}_{\text{avail}}(e) > \text{bw}_{\text{requ}}(n_p^s, \hat{n}^S)\}$;
9 $e_{n_p^s,\hat{n}^S} \longleftarrow$ dijkstra_shortest_path$(n_p^s, \hat{n}^S, G_{n_p^s,\hat{n}^S})$;
10 **if** $\forall n_p^s \in N_{\hat{n}^S}, \exists e_{n_p^s,\hat{n}^S}$ **then**
11 set_mapped$_{\mathscr{M}}(\hat{n}^S)$;
12 **if** $N_{avail} = \emptyset$ **then**
13 return error;
14 return $\mathscr{M}$;

5.3.2.2 Genetic Embedding Algorithm

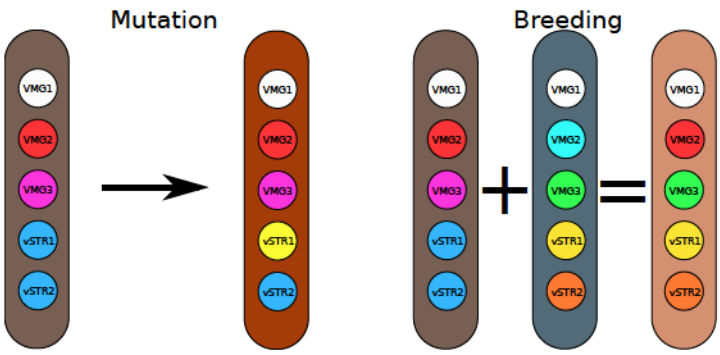


Figure 5.9: Illustration of our mapping "genotypes", mutation and breeding procedures.

A Genetic Algorithm (GA) [Srinivas and Patnaik, 1994] is a meta-heuristic that considers a *population* of candidate solutions, each of which having a *genotype* encoding its characteristics. Through an iterative process called *evolution*, the population is used to create (or *breed*) a new *generation* of solution by combining the genotypes of individual from the previous generation. During the breeding, genotypes can be *mutated* so that new genes

Algorithm 2: Genetic algorithm used for optimization

```
   Data: Service Graph G^S = (N^S, E^S), Physical Network Graph G = (N, E)
   Result: A Mapping M
1  pool ← Generate 50 random valid mapping from G^S and G;
2  while timeout is not reached
3      AND iteration < max_iteration
4      AND score_improvement > ε do
5      pool ← sort_by_fitness(pool) ;
6      while children_pool is not full do
7          parent1, parent2 ← weighed random draw from pool ;
8          children_pool.add( breed(parent1, parent2)) ;
9      pool ← ∅ ;
10     for each individual i in children_pool do
11         mutate(i) ;
12         M ← dummy_algorithm(G^S, G, i);
13         pool.add(M);
14     iteration ← iteration + 1 ;
15     score_improvement←stallion/pick_best(pool);
16     stallion ← pick_best(pool);
```

can enter the genetic pool. One can compute the *fitness* measure (the value of the objective function for each solution) which will be used to stochastically select the best individuals to breed from for each iteration. GAs have already been used to solve CDN allocation problems in [Bouten et al., 2015]. However, authors only considered the placement of caches without selecting pre-configured routes. [Qu et al., 2016] also used a GA to tackle Scheduling and Resource Optimization in NFV, with a different objective of reducing the schedule makespan.

For our problem, we consider that the initial population is a set of valid mapping computed by running several times the dummy algorithm. Each solution possess a genotype consisting of the node mapping (e.g., vMG 1 → Node N°5), which can be subject to mutation and will be used to "breed" new solutions, as show in Figure 5.9. The fitness function is the cost function associated to the mapping of the solution, taking into account network and system costs.

Listing 2 shows our implementation of the genetic algorithm. We start by generating 50 valid solutions for our mapping problem (line 1). Then, each mapping is sorted by fitness (i.e., decreasing cost function) and associated with a weight corresponding to the inverse of the objective function. The children pool (line 7-8) is built by drawing the parents from the parent pool from a weighted uniform distribution, and apply the breeding process. The

Table 5.3: # of Services graphs and computation time

	Full Class		**Reduced Class**		
	Geant		Geant	R^{20}_{58}	R^{40}_{115}
	unfiltered	filtered			
gen. time	20 s	206 s	10.1 s	10.2 s	52.1 s
# services	35856	538	10	10	10
embedding time (ILP)	-	563 s	15.0 s	14.8 s	96.0 s

Table 5.4: Cost Parameters

Resource Type	**vCPU**	**RAM**	**Storage**	**Ratio vm/Mbps**	**Equivalence**	**price per m**
vMG	2	4GB	20GB	1 VM for 1,250 Mbps	m4.large	72$
vSTREAMER	36	244GB	48TB	1 VM for 5,000 Mbps	d2.8xlarge	3,974$
Network						0.63$/Mbps

breeding process is a uniform random selection of genes from each parent. Then, for each children, we perform a random mutation (some mapping can be changed to another node from the physical network) (line 10-13). Finally, we compute the mapping for each children and move on to the next iteration. The algorithm stops after a timeout or a maximum number of iteration or when the score is not improving anymore for a long period of time (line 2-4).

5.3.3 Service Graph Generation and Embedding Evaluation

5.3.3.1 ILP-based embedding Evaluation

In this section, we evaluate the Service Generation Heuristic and compare it with the exhaustive method. To do so, we developed a simulation environment that generates services according to random SLAs, optimizes them with our ILP solver and finally selects the best embedding. We carried out our simulations on a real topology retrieved from an open database [Knight et al., 2011]. We particularly ran our tests on the Geant topology (consisting of 28 nodes and 35 edges), as it contains the most complete set of data (Bandwidth, GPS coordinates, etc.) available. The topology is loaded and NFVI-POP are assigned a capacity of 300 Virtual Machines. Each vMG and vStreamer is dimensioned according to the served bandwidth, based on lab measurements on our prototype implementation conducted within the T-NOVA project [T-NOVA Consortium, 2016b]. For each SLA, CG and CDN nodes mapping are selected randomly, based on weighted uniform draws on all the nodes in the graph. We computed weights as the total amount of edge bandwidth for each node, selecting proportionally the most weighted nodes for the CDN peering points (reflecting the fact that those nodes are usually well connected) and the less weighted nodes for the CG

(nodes located at the edge of the network). In this section, we first start by comparing the time required to generate the services of the Full and Reduced Service Graph Classes, and subsequently we compare the prices obtained by embedding the services on the physical topology.

5.3.3.2 Service Graph Generation

Table 5.3 shows the computation time results that was obtained on a Laptop equipped with an Intel i7-4600U CPU @ 2.10GHz, after generating the services graphs for a SLA composed of 4 CGs and 2 CDNs peering points. We compare the generation time for both the Full and the Reduced Service Graph Classes. For the Reduced Class, we generated Erdos-Renyi random graphs R_e^n with n nodes and e edges.

Most of the computation time is spent on generating the service graphs and computing the service mappings. For the Full Class, services can be obtained very quickly despite the fact that the tree algorithm generates 30k+ services. Filtering the service-isomorph graphs helps reducing this number down to 538, albeit with a significant increase in computation time. This can be explained by the fact that the graph isomorphism algorithm we use, VF2, has a very high $\mathcal{O}(V!V)$ worst case complexity, with V denoting the number of service edges. In contrast, applying the heuristic reduces the number of possible services drastically down to 10. It should be noted that the computing time of the heuristic increases with the size of the physical topology, contrary to the exhaustive method used to compute the Full Class.

Finally, when comparing the total embedding time, we can see that using the heuristic reduces the generation time for the optimal solution by a factor of 37. We can also see that the size of the graph greatly impacts performances: computing the embedding on R_{115}^{40} is about 5 times longer than on R_{58}^{20}. We will see in Section 5.3.3.4 that the genetic algorithm can address scaling issues by providing approximates.

5.3.3.3 Heuristic vs exhaustive method embedding quality comparison

For the experimentation, we used the SCIP toolkit to generate mappings[3]. SCIP employs a branch-and-bound approach, which is complemented with linear programming relaxation and cutting plane separators, using constraint specific domain propagation algorithms and conflict analysis [Achterberg, 2009].

Using the services generated in the Full and Reduced class, we conducted an experiment on 20 randomly generated SLA with up to 4 CG and up to 2 CDN. Table 5.4 summarizes the cost parameters. Our simulation results are shown in Figure 5.10. For comparison, we also included trivial services : (A) with 1 vMG and 1 vStreamer and (B) with 4 vMGs and 4 vStreamers. The results obtained by optimizing on the Reduced Service Graph Class are very close to the results obtained by testing every graph in the Full Service Graph Class. In fact, we see a very small 0.71% cost increase, due to suboptimal service edge being selected

[3] http://scip.zib.de/

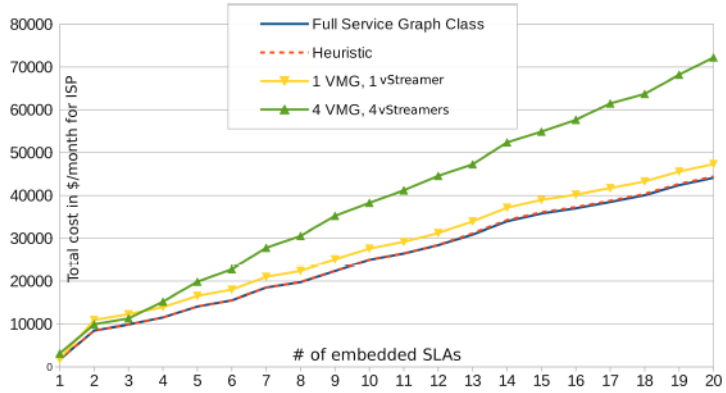


Figure 5.10: Cost Comparison of serie of successive embedding performed on the Geant Topology

Table 5.5: Genetic Algorithm parameters

Parameter	**values**
Pool Size	50
Timeout	5 s
Mutation Rate	30%
Minimum/Maximum iterations	5/20
Minimum Score Improvement	at least 1% for 3 consecutive iterations

by the heuristic. The heuristic has also lower cost than trivial service (A) by 7.4% and service (B) by 63.8%. This shows that selecting the best service, in the possible service graphs, is very important to have the best mapping.

Let us now proceed with the evaluation of the genetic algorithm that aims at improving the scalability of the vCDN embedding problem.

5.3.3.4 Genetic Algorithm

The previous section showed that the impact of using the heuristic for service generation is very small. However, optimizing the embedding of the vCDN service over large network is impracticable, as the ILP convergence time is exponential. We evaluated how the genetic algorithm can be an alternative to the ILP for large networks.

5.3.3.4.1 Convergence Time

In Figure 5.11, we evaluate the convergence time of the dummy and genetic algorithms compared to the ILP. We used randomly generated SLAs with 4 CG and 3 potential CDN peering points. We ran the embedding on a synthetic Powerlaw generated physical network

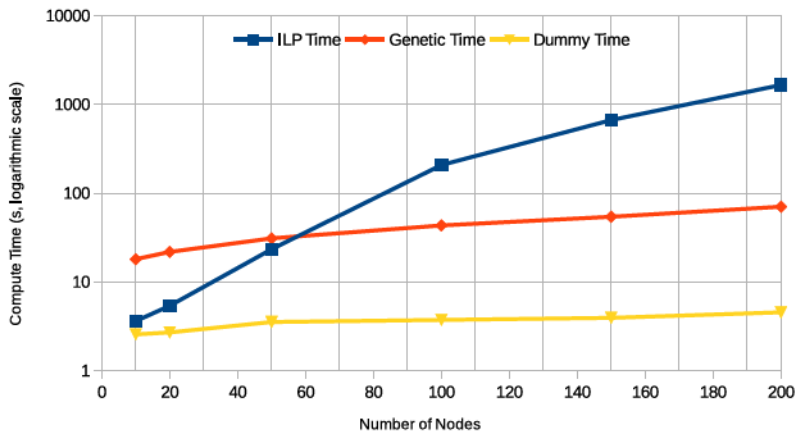


Figure 5.11: Comparison of solvers computation time

ranging from 10 to 200 nodes and repeated the experiment 50 times, showing the mean value. Table 5.5 shows the hand-tuned parameters for the genetic algorithm.

The results prove that the ILP runs faster that the genetic algorithm for small networks (from 10 to 50 nodes). Indeed, the genetic algorithm takes at least 5 iterations to generate a solution, which especially impacts the convergence time for small-sized networks. After 50 nodes, the convergence time of the ILP explodes, while the genetic algorithm has a more linear trend. For example, for 200 nodes, using the genetic algorithm reduces the computation time by a factor of 10. Finally, the Dummy algorithm shows an even smaller convergence time increase with the network size. Even if it does not produce high quality results, it can be used to provide a feasible solution quickly.

5.3.3.4.2 Embedding Quality

Keeping the same experimental settings, we computed the objective function for each physical network size, and compared the results for the three candidate algorithms. Figure 5.12 shows how the ILP, dummy and genetic algorithm compete when considering embedding quality in term of cost minimization for the ISP.

We can see that the ILP has always the smallest score (i.e., the best embedding), but the genetic algorithm stays within a 20% cost increase, which is acceptable given the saving on computation time for large physical graph.

5.3.3.5 Toward an adaptive embedding algorithm

We saw that both heuristic and exhaustive service graph generation procedure, on one hand, and ILP and genetic algorithms, on the other hand, offer interesting properties:

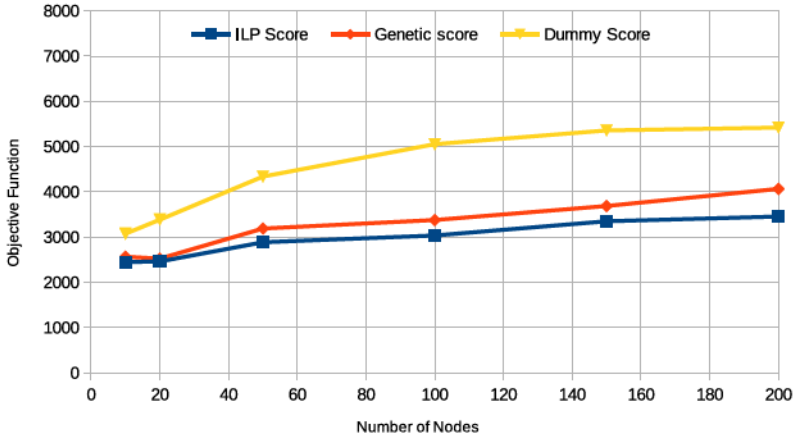


Figure 5.12: Comparison of solvers performance

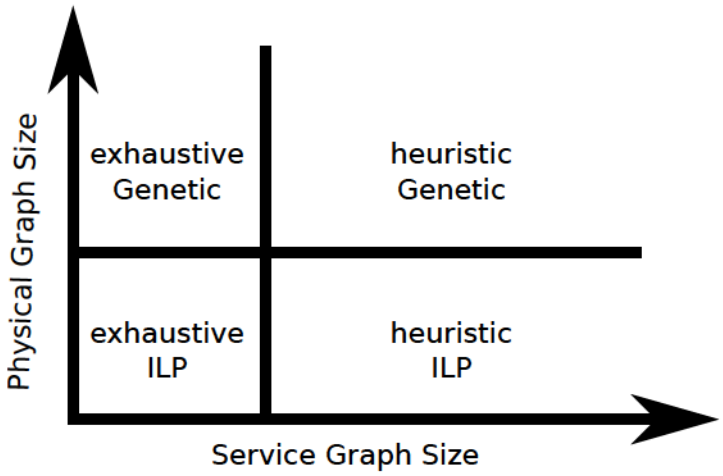


Figure 5.13: Adaptive strategy selection algorithm

- The exhaustive generation procedure, while assuring that every possible service graph is evaluated, offer the best embedding, but is very sensitive to the size of the service graph.
- The heuristic reduces the number of service graph to evaluate, but offer slightly degraded solutions.
- The ILP offers the best possible solutions but does not scale with the size of the physical network.
- The Genetic algorithm offers better convergence time for large physical graphs, but

produces suboptimal solutions wrt the ILP. It is also slower for small physical networks.

We consolidated the four options into a unique adaptive algorithm that selects the best one according to the characteristics of the service graph and the physical network, as illustrated on Figure 5.13. In our implementation, the detection of the 2 thresholds (1 for service graphs and 1 for physical network) is done manually by trial and error for each physical graph. We plan to implement a robust threshold detection mechanism in future works.

5.4 CDNaaS Dynamic SLAs support

The previous section detailed algorithms that derive Service Graphs from vCDN SLAs and generate a mapping between the ISP physical network and the vCDN service implemented as a Service Function Chain. Those algorithms, however, considered only static SLAs where the bandwidth demand and CG location stay the same throughout the SLA. However, this hypothesis does not hold as most of the CPs experience time-varying connectivity needs which can be due to the typical diurnal pattern of content consumption, planned release of highly anticipated series episodes[4], or flash crowds.

In this section, we leverage on the previous optimization algorithms and further extend them to support dynamic SLA that can benefit the vCDN Customers and the ISPs. First, we present a dataset used to simulate dynamic distributed traffic consumption. Second, we discuss the steps required to deploy and operate a virtual CDN deployed on an ISP's network. Furthermore, we present evaluation results of the proposed solution, based on simple models. Lastly, we elaborate on operational parameters that are used to further optimize the solution.

5.4.1 vCDN steps of deployment and operation

We show the five major steps required to deploy and run a virtual CDN in Figure 5.14, as follows:

1. The vCDN customer collects historical data.
2. The vCDN customer uses the data to predict future bandwidth consumption.
3. The vCDN customer uses the traffic predictions to generate SLAs that formalize the contract with the ISP.

[4]HBO GO, the online Video on Demand (VOD) service airing Game of Thrones episodes suffered outage during the premiere of Season 7 https://goo.gl/JK6RAZ

4. The ISP prices each SLA and embeds the vCDN as a service function chain into its network.

5. Because connectivity demand is very dynamic, the ISP further optimizes services to cope with the fragmentation of its components to reduce costs.

These five steps are discussed, in more detail, in the following.

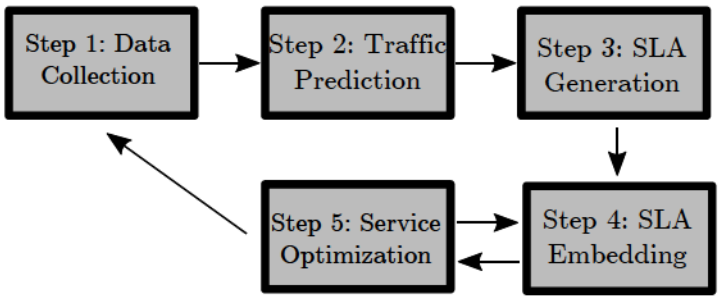


Figure 5.14: VCDN five steps of deployment and operation

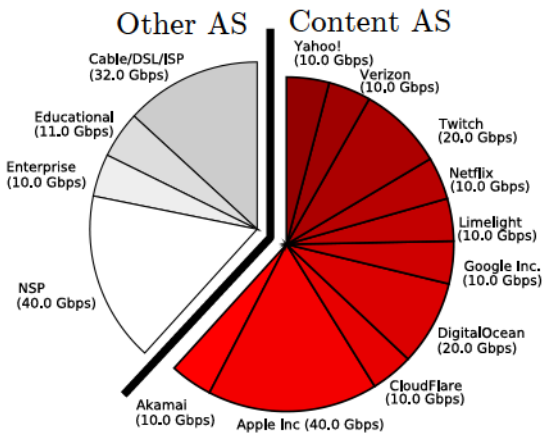


Figure 5.15: Step 1 – Data collection showing a typical mixture of ASs at a peering point

5.4.2 CDN traffic estimates

Collecting historical data is required for the vCDN customer to understand its traffic volume and to provision servers and connectivity. In parallel, the ISP must consider its VNF characteristics to determine how many servers will be used to host the software components.

Traffic data is not publicly available; however, reasonable estimates can be inferred from Internet Exchange Points (IXPs). IXPs consist of infrastructures where the ISP and CDN usually establish peering between their autonomous systems. We extracted publicly available

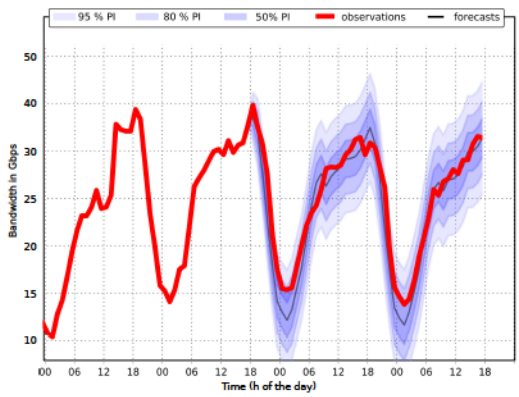
daily statistics of more than 50 peering points from the service monitoring webpages of four IXPs [5]. The traffic of each peering point aggregates its specific mixture of content, enterprise network, and ISP autonomous systems. Using the data from the peeringdb.com API, we can precisely estimate the bandwidth allocated to each autonomous system. We extracted the business type of each autonomous system: "cable/DSL /ISP," "Network Service Provider (NSP)," "education/research," "enterprise," "nonprofit," or "content" (CDNs and large CPs). The "content" autonomous systems represent a very significant part of the traffic, as shown in Figure 5.15, so we calculated this traffic by dividing the total traffic by the autonomous system bandwidth share.

5.4.3 Traffic Prediction

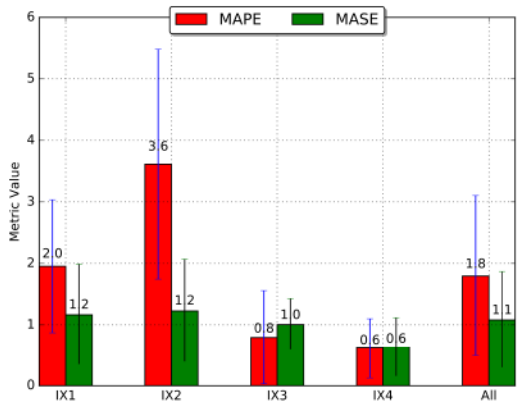
For traffic prediction, we go beyond related work that relies on the prediction of resources [Kryftis et al., 2016] by also taking into account changes in traffic demands. Traffic in CDNs — and VoD in particular — follows a very specific diurnal pattern, as Figure 5.16a shows. The patterns are also specific to geographical regions, due to time differences and the broadcasting of local content (such as a regional sporting event). In addition, exogenous behavior might occur — for example, when the first release of a popular TV program dramatically increases data consumption. Such events cannot be incorporated into the model by relying on historical data only. For traffic prediction, we used a Seasonal Autoregressive Integrated Moving Average (SARIMA) model, which is well suited to the prediction of aggregated and backbone traffic [Sang and Li, 2002] and can generate forecasts easily. We also computed prediction intervals to adjust the predicted traffic level according to model fitness. We focused on measuring the accuracy of forecasts performed on a 48-hour window. To this end, we conducted the evaluation using a 50+ peering points dataset belonging to four different IXPs (IXP 1 to IXP 4), collected over three weeks. We used two metrics for the evaluation: the Mean Absolute Percentage Error (MAPE), an easy to understand percentage measure of the accuracy of forecasts, and the Mean Absolute Scaled Error (MASE), a more advanced measure that compares the performance of the forecast with regard to a naive prediction [Hyndman and Koehler, 2006] (depicted in Figure 5.16b).

Based on MAPE, we observe that the models performed rather well with a 1.80 percent mean error on the whole dataset. The MASE metric shows mean scores around 1.1, with high standard deviations, meaning that depending on the considered time series, the proposed model might be well adapted in some cases but not others. Weaker results are due to technical maintenance occurring close to the prediction windows for IX1, as well as trend breaks on the peering-point data caused by an autonomous system being added or removed from IX2. (Handling such planned events is possible with exogenous variables but this is out of the scope here). After predicting the traffic demand, the next step is to have the ISP generate the SLA.

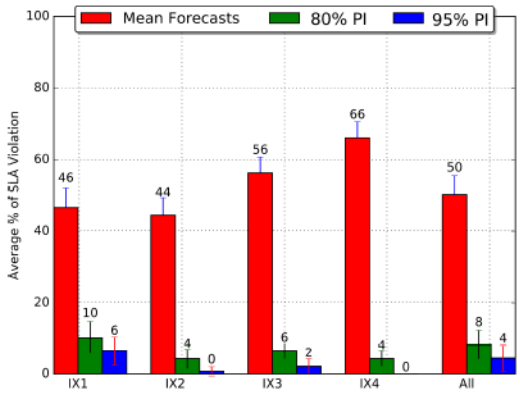
[5] The dataset is availabe at https://data.nextnet.top



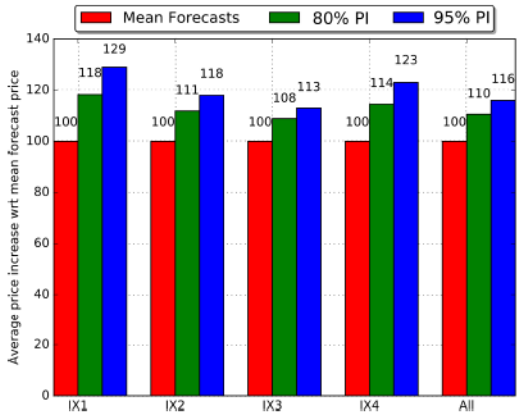
(a) Forecast and prediction intervals



(b) Forecast quality metrics



(c) Ratios of SLA Violations



(d) Prices wrt prediction intervals

Figure 5.16: Step 2 – Forecast, SLA and Discretization evaluations

5.4.4 SLA Generation

A vCDN SLA provides the assurance that a certain amount of bandwidth will be available to a given geographic area over a given time period. SLAs also contain two necessary specifications: the maximum acceptable delay and the list of existing peering points, where legacy CDN traffic exits" the ISP network. Jeroen Famaey and his colleagues have presented a framework that exploits SLAs to create content delivery federations between the ISP (for the network), cloud Providers (for storage), and CPs (for purchasing the service)[Famaey et al., 2012b]. We follow a similar approach, albeit considering the ISP as the only actor providing the service. CDN Providers generate SLAs following three steps:

- The predictions $\hat{t}$ are discretized to generate a step-like envelope of constant bandwidth demand chunks, as shown in Figure 5.17(a). The algorithm filters them accord-

ing to a smoothing window of length w and applies a K-Mean method, resulting in a discrete step-function with c possible values.

- The SLAs are generated by creating layers of demand chunks that have the same bandwidth value, thanks to the function $D(\hat{t}, w, s)$ (see Figure 5.18).
- The ISP applies its p pricing function to compute the right SLA price p_{SLAs} to be proposed to the CDN provider, according to the following equation:

$$p_{SLAs} = p\big[D(\hat{t}, w, c))\big] \tag{5.10}$$

By definition, forecasts produce the closest predictions possible, whereas the CDN needs a buffer to avoid underestimating its traffic. This buffer can be obtained using prediction intervals as a basis for SLA creation. Selecting a high prediction interval reduces SLA violations. As shown in Figure 5.16c(c), choosing an 80 percent or 95 percent prediction interval decreases the SLA violation rate to 8.2 or 4.3 percent, respectively. The strong advantage of using prediction intervals is that they automatically adapt to the model's predictive power. If data is not fitted well, the model generates large prediction intervals. The drawback of overestimating the forecast is that it leads to an increase in the resulting SLA prices, as the level of bandwidth provisioned for each SLA increases. This, in turn, affects the total pricing for the CDN Provider, as shown in Figure 5d. Choosing the 80 or 95 percent prediction interval models increase the price by 10.5 or 16.1 percent, respectively. From that point on, the CDN Provider decides the acceptable risk/price ratio with respect to the expected quality. The ratio is intended to be higher for sponsored content (where the content is paid by ads, so a quality drop can be tolerated) than for premium content (where the content is paid by the user, with no compromise on quality).

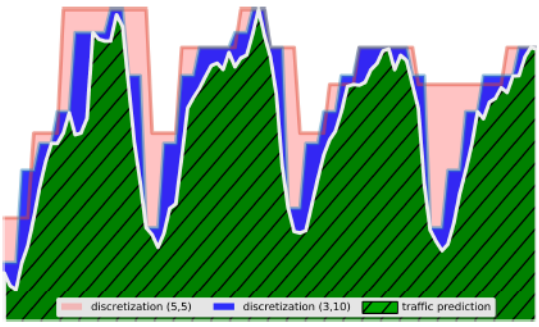


Figure 5.17: Different discretization parameters

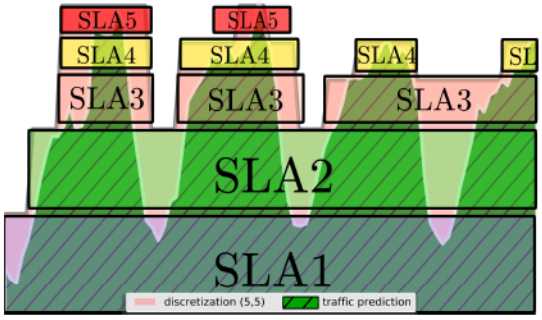


Figure 5.18: SLA generated from discretized predictions

Figure 5.19: Step 3 – Discretization and SLA Generation

5.4.5 SLA Embedding

For this step, we directly use the adaptive embedding algorithms presented in Section 5.3. Since the evaluation is performed on a small-sized network (Geant with less than 50 nodes), the ILP strategy is used for the embedding. For service graph generation, the selection depends on the number of n_{cg} CGs and n_{cdn} CDN present in the SLA. In this case, the heuristic service graph generation strategy is used when $n_{cg} + n_{cdn} > 3$.

5.4.6 Service Scheduling

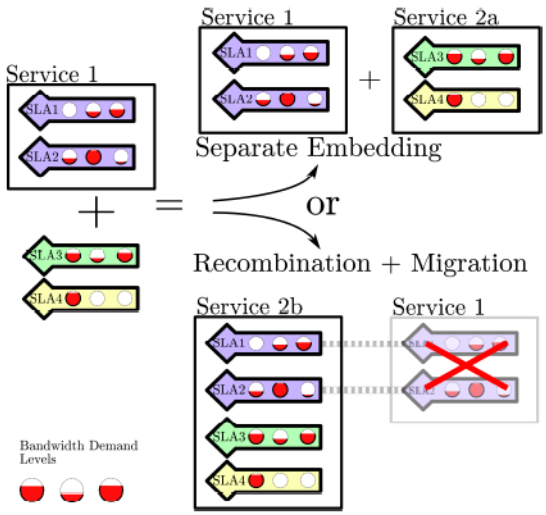


Figure 5.20: Step 5 – Dynamic cost-aware scheduling

We hereby present two optimization methods to cut the cost on the ISP side. We first elaborate on the possible service combinations, before studying the impact of the ISP pricing function on the SLA generated by the CDN.

5.4.6.1 Service Migration

Over time, new SLAs might arrive or existing SLAs might expire, resulting in changes to the underlying services deployed by the ISP. If an SLA expires, the ISP releases the bandwidth reservations for the virtual links associated with that service. On the other hand, when new SLAs arrive, the bandwidth demand increases. In this case, it might not be possible to embed the incoming SLAs using the same physical links or the same PoP, due to capacity constraints. To alleviate this, we consider two strategies available to the ISP to minimize the embedding cost (see Figure 5.20).

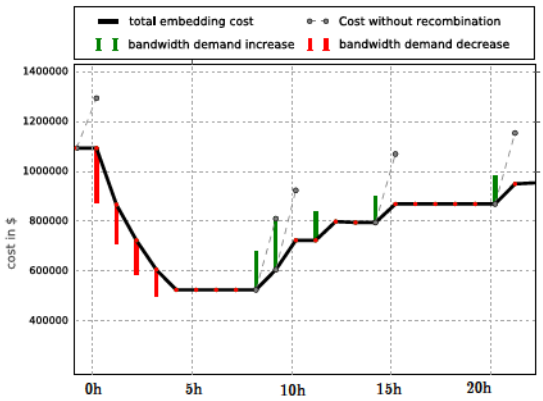


Figure 5.21: Example of the evolution of the cost of Service Embedding with SLA generated from 4 ISPs, 4 legacy CDN peering points over 24h

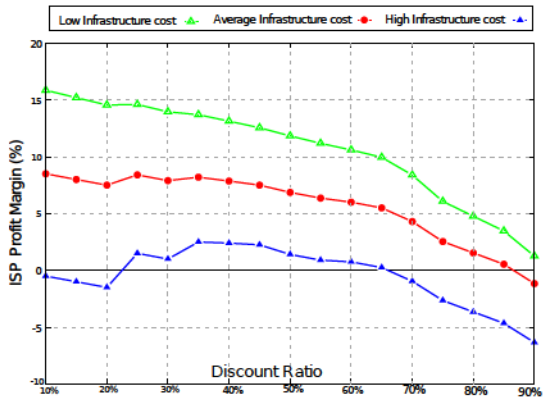


Figure 5.22: Evaluation impact of service optimization and discount policy

- **Strategy 1 — Separate embedding**. The ISP creates two services side by side, with each service being in charge of implementing a subset of the SLAs required by the CDN.
- **Strategy 2 — Recombination+migration**. The ISP recombines all of the SLAs within the same service. Even though it might appear more efficient for resource spreading, VNF migrations might be necessary, increasing the cost of the new service.

The ISP selects the strategy that minimizes the embedding cost and subsequently deploys the new services, whenever there are new or expiring SLAs.

Figure 5.21 shows the evolution of the embedding cost over time. In particular, the figure highlights the variation of the optimal cost, bandwidth demand increases and decreases, and the cost savings induced by Strategy 2 of recombining services (including migration costs).

5.4.6.2 Adjusting the ISP pricing scheme

The migration of VNFs is an opportunity to further optimize the service, but also a necessity to reduce service fragmentation. In this respect, we studied how to reduce service fragmentation through the impact of the ISP pricing function on the SLAs.

ISPs and CDN Providers have conflicting interests regarding SLAs. On the one hand, the CDN Provider wants to generate SLAs as close as possible" to its traffic forecast to limit the overhead of the discretization algorithm. The best strategy for the CDN would be to generate a new SLA for each forecast value, precisely matching the estimates. However, this strategy would lead to a large number of SLAs and, in turn, the need for new service

embeddings, resulting in substantial migration costs for the ISP. The ISP thus wants to limit the number of SLAs to a bare minimum. To this end, the ISP can promote fewer and longer-lasting SLAs through its pricing schemes. By applying a discount rate depending on the duration of the SLAs, the ISP can incentivize the CDN to adapt its discretization strategy to reduce the number of SLAs.

To understand how the discount ratio influences ISP revenue, we considered an exponential-decaying pricing function for SLAs up to 24 hours and assumed that the CDN would generate the SLAs by choosing the discretization parameters that minimizes its cost with regard to its traffic predictions. SLAs were then used to run a simulation and compute the overall embedding cost for the ISP. Finally, we computed the ISP profit from the ISP cost and the CDN cost and repeated this experiment for several discount ratios and cost hypotheses.

Figure 5.22 shows the results we obtained. We considered three hypotheses depending on the cost structure of the ISP (low, average, or high), taking into account VM hosting, networking, and VM migration costs. The main outcome is that the cost structure of the ISP influences the relevance of implementing the discount mechanism. In our hypothesis, for the high cost structure, we see that the discount mechanism must be implemented. In fact, the service is lucrative only for discount ratios between 25 and 65 percent — with the highest profit margin value being approximately 40 percent. At the same time, for the average cost structure, we see that the discount mechanism has little impact on the ISP revenue for values below 35 percent. The ISP can therefore use the discount mechanism to adjust its offer to the market price for CDN service, without hurting its profit. Finally, for the low cost structure, we see that implementing the discount mechanism is not effective, because it only reduces the profits.

To conclude, the discount mechanism should be adjusted for each ISP cost structure and can decrease Operational Expenditure (OPEX) by influencing decision mechanisms for hardware and software investments [Seaman et al., 2012].

5.5 Conclusion

The deployment and operation of vCDN, as reported in this section, would bring strong added value to the wide range of use cases, as follows:

- Footprint extension in low density areas letting a CDN expand its connectivity in an area where it is not economically sustainable to set up permanent peering.
- Quality improvement for niche markets letting a CDN deliver high-quality media (4K, 8K...) over HTTP with only a small delay toward the streaming server.
- Bandwidth burst—letting a CDN temporarily increase its bandwidth in a geographical area to serve more users.

- Virtual CDN operator letting ISPs deliver the content, making it possible to experiment with novel content distribution paradigms through virtualization.

In particular, for the case of footprint extension, one of the missions of the FEDER European development program is to fight against numerical deserts" by injecting substantial amounts of money to strengthen access to ICT in rural areas. The ISP has a role to play, not only in bringing connectivity to these areas but also in proposing efficient content availability at all levels, including those that are environmentally friendly thanks to virtualized vCDN platforms. The ease of deployment, operation, and optimization, provided by our proposal's SLA generation, SLA embedding, and service optimization (steps 3–5), should contribute to the appropriation of such use cases by the ISP.

For the case of quality improvement for niche markets, in terms of ultra-high-definition (UHD) instant streaming services over HTTP, maximum delays needed to reach the 100 or 200 Mbps mark are one order of magnitude greater than the delays typically experienced today. Integrating delivery solutions within ISP networks through a vCDN deployment will greatly help CPs to target this UHD niche market. Indeed, thanks to our proposal's traffic prediction, SLA generation, and SLA optimization (steps 2, 3, and 5), prediction and assurance of end-to-end latencies are possible.

For the bandwidth burst case, because CPs still face outages when releasing their highly anticipated content to the market, vCDN could help by offloading their own networks by pushing their content even deeper into the ISP network. Such operation could also offer guaranteed results through traffic prediction and SLA generation (steps 2 and 3 in our proposal). On top of offering lower costs compared to traditional physical CDN deployments, this model can provide service quality guarantees.

Finally, for the case of the vCDN operator, virtualizing content delivery can ease experimenting with novel architectures and algorithms, such as content-centric networks and multisource delivery protocols. In future work, we plan to quantitatively assess the gain for every actor (CP, CDN, and ISP) of the vCDN.

CHAPTER

6 Conclusion and perspectives

Over the years, the Internet has always found ways to reinvent itself to adapt gracefully to new services and new use-cases and commercial interests play a major role in shaping its ongoing development. Today, with the democratization of online videos, massive content distribution is the latest challenge that needs to be addressed. Only this time, on top of technical difficulties, actors must overcome conflicting economic interests and fluctuating regulations at unprecedented scale. Without a sound collaboration between them, achieving the target of providing the expected quality of experience to a worldwide audience of binge-watching customers would contribute to the ossification of the Internet, by multiplying the deployment of parallel networks of edge caches.

This thesis demonstrated that Software Defined Networking and Network Function Virtualization, the two pillars of Network Softwarization, can be used to implement a versatile virtualized edge network capable of delivering content according to high-level SLAs. With NFV-capable points of presence, a third party can host its own implementation of the Content Delivery function within the ISP edge to accommodate its specific needs. In doing so, the said it is able to complement its own existing network with guaranteed bandwidth and latency to offer unprecedented Quality of Experience to its customer. Finally, SDN was used to implement network management and Service Function Chaining in the NFV infrastructure.

Going one step further than traditional Cloud Computing models, the content delivery network as a virtual network function, CDNaaS, makes collaboration not only feasible, but also mutually beneficial by reducing the need for peering and by creating a new revenue streams for ISPs. Thanks to high-level Service Level Agreements, ISPs can propose the CDN function as a service while optimizing its deployment to target cost reduction. In this thesis, we proposed several algorithms specifically designed to make cost reduction and dynamic scheduling of SLAs possible.

Even if making the collaboration possible between content delivery actors provides better quality for the End-Users, it does not necessarily promote competition on the Content Provider market. Promoting competition is important to prevent vertical integration of content delivery that has the same consequences as the lack of collaboration wrt to the ossification of the Internet. On the other hand, End-users willing to access a specific content do not necessarily care about the collaboration scheme implemented behind the hood by the actors of the value chain, as long as they get the best price and best quality. Based on this observation, we designed a decentralized brokering mechanism that selects, for each content session, the best Service Chain for delivery, by computing the optimal mix between CPs and CDN providers (including CDNaaS) through the use of blockchain and smart contracts.

We believe that our work is an important step toward practical virtualization of the content delivery. However, our proposed architecture and models are first steps in this area with exciting prospects for future work that follow naturally from this thesis.

Enhancing Content distribution through fog-scale storage

In Chapter 3, we described the integration of a distributed storage solution based on an object storage engine relying on large storage arrays. Even if this technology has been widely adopted by cloud providers for a decade, it tackles naively the problems of unreliable latency, lack of mobility support and location-awareness.

The concept of fog computing promotes the usage of edge resources in a cooperative manner. For content distribution, resource pooling allows integrating the communication resources of multiple edge devices and exploiting the diversity of users' channel conditions [Tang et al., 2017].

The user-centric collaboration model proposed in Section 4.3 would benefit for such an approach by providing additional reliability guarantees. The difficulty of developing applications that are both correct and efficient in usage (for both an economic and performance perspective) of large-scale distributed resources at *Fog-computing scale* is an interesting research challenge to be addressed in the near future.

Net Neutrality and access to culture

In Chapter 4, we quickly discussed the Net Neutrality questions that the collaboration between ISPs and CPs raise. Thoroughly addressing this topic goes far beyond the scope of this thesis and requires a specialized juridical opinion, especially in a context when the Net Neutrality regulations are quickly evolving.

This matter also relates to the broader topic of public welfare. Preserving the equitable access to a broad culture is of paramount importance when considering that most of the big OTT players come primarily from the United States and participate in the diffusion of a mono-culture. If a handful of US content providers own the studios that produce the

content, the networks that distribute the contents and the search engines or recommender systems to discover the content, then there would be no alternative left to the End-Users. Some national regulations (such as the French cultural exception political concept) promote a state intervention in the means of production and diffusion of cultural goods to assure the subsistence of a sovereign culture/media industry. Such regulations might not adapt fast enough to cover new content distribution paradigms.

For these reasons, we plan to carry out a multidisciplinary study that:

- evaluates collaborative content distribution technologies wrt their legal validity for Net Neutrality regulations;
- questions the economic models for the whole value chain to highlight uncompetitive behaviors;
- proposes an extension to existing legislation to take into account new content distribution paradigms.

Reliability and truthfulness of Blockchain and Smart Contracts

In Section 4.3.2.1, we presented a blockchain-based model that leverages smart contracts that compute the best collaboration scheme for a content session. This model suffers from a legal uncertainty regarding the exact status of smart contracts. Some argue that they are based on a thin conception of what law does, and how it does it [Levy, 2017].

Another aspect that may impede the adoption of such smart-contract based schemes is the impossibility to appeal the result their results. Indeed, without a proper formal verification of the code, the implementation may not be free of errors and can be exploited. At best, this reduces the trust users have in the blockchain system (defeating its purpose), and at worst it can destroy value by writing spurious results in the blockchain forever [Atzei et al., 2017].

For these reasons, we plan to propose a model to improve both legality and trust in smart contracts by using an external (and possibly human) legal arbiter. This study will also be conducted in a multidisciplinary approach with legal experts.

We hope to investigate these topics as part of our ongoing efforts toward shaping the future of content distribution on the Internet.

APPENDIX

A Résumé en Français

L'objectif de cette thèse est de prouver qu'il est possible, en utilisant les nouvelles technologies de programmabilité de réseaux de créer un éco-systèmes où la distribution de valeur liée à la diffusion de contenu par contournement (OTT) est mieux répartie entre les différents acteurs de la chaine de valeur.

A.1 Motivation pour la virtualisation du CDN

Nous illustrons dans un premier temps quels sont les problèmes actuels de manque d'équilibre dans la chaine de valeur, pour les fournisseurs d'accès à internet (ISP, Internet Service Providers) qui doivent assurer la majeur partie des investissements dans les infrastructures réseaux. Ces infrastructures sont en majorité utilisées par les fournisseurs de contenus (CP, Content providers) pour livrer le contenu multimédia demandé par les utilisateurs, soit directement, soit par des intermédiaires techniques spécialisés, tels de les CDN (Content Delivery Networks).

Notre travail est motivé par le fait que (1) les ISPs ne bénéficient que très peu de la manne financière générée par l'augmentation de la popularité des services par contournement et que (2) les utilisateurs n'ont pas la possibilité de bénéficier d'un marché de la livraison de contenu très concurrentiel, car les acteurs majeurs ont tendance à concentrer les rôles en bloquant les "locus" de compétition très en amont de la chaine de valeur.

Les ISPs ne pouvant pas augmenter le revenu qu'ils tirent des opérateurs en raison de la commoditisation du marché de la connectivité IP et que les règles de neutralité de réseaux empêchent ces derniers de créer un marché biface où les fournisseurs de contenus seraient facturés.

Nous défendons l'idée selon laquelle la virtualisation de la fonction de livraison réseau permet aux ISP de proposer des services similaires au cloud-providers tout en assurant la

qualité de livraison bout en bout. Nous arguons de même qu'en plus des possibilités offertes par l'approche NFV, des efforts supplémentaires sont nécessaires, comme la séparation des intérêts entre les acteurs implémentant la couche physique et les acteurs responsables de l'utilisation de la couche virtualisée. Dans cette vision, nous proposons CDNaaS, précisons les aspects de l'architecture réseau nécessaire à une collaboration mutuellement bénéfique entre tous les acteurs et détaillons les problématiques spécifiques liées au plongement de cette fonction réseau virtualisée dans le réseau opérateur.

A.2 CDNaaS: l'Implémentation de référence

Notre première contribution consiste à produire l'architecture et l'une implémentation de référence d'une Fonction Réseaux Virtualisée (VNF virtual Network Function) de Livraison de contenu (vCDN, virtual Content Delivery Network): CDNaaS (CDN as a Service). Reprenant les principales fonctionnalités offertes par les CDNs traditionels, CDNaaS gère l'ingestion de contenu, le placement de contenu dans les caches situées en bordure du réseau, ainsi que l'assignation des caches pour chaque requête utilisateur.

Les fonctions sont assurées par 3 modules dont nous détaillons l'implémentation et évaluons les résultats en terme de transfert de données, passage à l'échelle et intégration avec l'architecture standard MANO proposée par l'ETSI. Nous avons effectué ces tests sur une véritable plateforme NFV déployée dans le cadre du projet T-NOVA.

A.3 Déploiement de CDNaaS dans le réseau opérateur

Notre deuxième contribution vise à démontrer que CDNaaS peut être vu comme une solution bénéfique à la fois pour les opérateurs réseaux télécom, mais également pour le reste de l'écosystème. A ce titre nous proposons une architecture qui vise à déployer des instances de CDNaaS développées par les CP ou CDN dans les réseaux opérateurs gérés par les ISP.

Pour ce faire, nous proposons le découpage du déploiement en 3 plans: (1) le plan infrastructure qui permet à l'ISP de garder confidentielles les données issues du déploiement physique des routeurs, liens et serveurs, (2) le plan de livraison de contenu qui permet au CP/CDN de déployer ses fonctions réseaux et de le gérer via des liens virtuels et (3) le plan de management où l'ISP et le CP/CDN négocient entre eux un SLA du haut niveau correspondant aux caractéristiques de la livraison de contenu (e.g. débit, latence, nombre de clients simultanés).

CDNaas vise à établir une collaboration entre ISP et CDN. Nous comparons CDNaas aux autres modèles de collaborations déjà existants à l'aide d'une modélisation basée sur la théorie des jeux. Après avoir noté de manière formelle les couts et dépenses des acteurs ciblés dans un tableau, nous écrivons le jeu en forme normale et cherchons à établir les

conditions permettant de rendre la solution CDNaaS optimale (i.e. où la stratégie CDNaaS est strictement dominante.)

A.4 Sessions de contenue centrées sur l'utilisateur

Notre troisième contribution étend la contribution précédente en mettant l'utilisateur au coeur du système de création de session de contenu multimédia. Nous proposons une architecture réseau basée sur la compétition qui permet à plusieurs acteurs de la chaine de valeur de proposer leurs services pour satisfaire une même requête utilisateur. Nous proposons un système de courtage basé sur le blockchain et les contrats intelligents, dans lequel les utilisateurs envoient leurs requêtes de contenus. Tous les CP étant en capacité de satisfaire cette requête (en licenciant le contenu à l'utilisateur) proposent leur service pour un prix donné en publiant à leur tour un contrat. De la même manière, chaque facilitateur technique (TE) hébergeant le contenu peut proposer ses services en publiant à leur tour un contrat intelligent correspondant. Le contrat intelligent initial arbitre le contrat le moins onéreux et la livraison de contenu peut commencer.

Nous décrivons les différentes modalités de déploiements des services réseaux de livraison de contenus résultants, basés sur les CDN traditionnels, des CDN virtuels (CDNaas) ainsi que des micro-CDN hébergés sur les passerelles réseau des utilisateurs finaux.

Nous évaluons le système ainsi proposé par une simulation à évènement discret qui montre que l'utilisation conjointe de différents facilitateurs techniques et fournisseurs de contenu réduit le prix final pour l'utilisateur. Nous avons également implémenté une preuve de concept à l'aide du projet hyperledger fabric montrant que les capacités de passage à l'échelle permettent d'assurer la création du contrat final dans les temps.

A.5 Allocation de ressource pour les VNF

Notre quatrième contribution considère le problème de l'allocation de ressources dans les NFVs appliquée à CDNaaS. Pour cela nous proposons une solution pour les trois sous-problèmes liés.

Pour le problème de la génération des chaines de services, nous proposons une heuristique permettant de réduire la taille de de l'échantillon des services possibles sans pour autant réduire la qualité du plongement final.

Pour le problème de plongement, nous proposons deux implémentations, basées l'une sur la programmation linéaire et pour l'autres nous utilisons une métaheuristique d'algorithmes génétiques permettant d'augmenter le passage à l'échelle de la solution linéaire pour les grands réseaux d'infrastructure.

Finalement pour le problème d'ordonnancement des VNFs, nous fournissons un dataset dérivé de l'observation longitudinale du trafic des points d'échange internet couplé avec

les données de peering. Grace à ce dataset, nous démontrons la possibilité de générer des prévisions endogènes du trafic. Nous proposons également un algorithme de discrétisation du trafic afin de générer les Accords de niveau de Services dynamiques, et montrons qu'il est possible de réorganiser les services précédemment déployés afin de les consolider en prenant en compte les coûts de migration des machines virtuelle. Nous concluons sur l'existence d'un modèle de facturation optimal par l'ISP pour adapter les choix de discrétisation du trafic aux possibilités de consolidation du service.

A.6 Conclusion

dans cette thèse nous avons montré que CDNaas peut être (1) une bonne réponse aux problématiques actuelles de livraison de contenu multimédia (2) mutuellement profitable pour les ISP, les CP et les utilisateurs (3) complémentaires aux technologies existantes de livraison de contenu pour réduire le prix et assurer la qualité et (4) gérable à l'aide d'algorithmes adaptés de génération de graphes de services et d'optimisation du plongement.

APPENDIX

B

List of publications

International Peer-reviewed Journal Articles

1. Nicolas Herbaut and D. Négru, "A Model For Collaborative Blockchain-based Video Delivery Relying On Advanced Network Services Chains," *IEEE Communication Magazine, vol. 55, issue 9, 09/2017.*

2. Nicolas Herbaut and D. Négru, D. Dietrich, and P. Papadimitriou, "Dynamic deployment and optimization of virtual content delivery networks," *IEEE Multimedia, vol. 24, issue 3, 07/2017.*

3. Y. Rebahi, M. S. Ghamsi, Nicolas Herbaut, D. Négru, P. M. Comi, P. S. Crosta, P. Lorenz, E. Pallis, and E. Markakis, "Virtual network functions deployment between business expectations and technical challenges: The t-nova approach," *Recent Advances in Communications and Networking Technology, vol. 5, 10/2016.*

4. Nicolas Herbaut, G. Xilouris, and D. Négru, "The surrogate vnf approach for content distribution," *COMSOC Multimedia Communications Technical Committee E-Letter,, vol. 10,p. 4–6, 2015.*

International Peer-reviewed Conference Papers

1. Nicolas Herbaut, Daniel Négru, David Dietrich, and Panagiotis Papadimitriou. "Service Chain Modeling and Embedding for NFV-based Content Delivery." *In IEEE International Conference on Communications (ICC). 2017.*

2. Nicolas Herbaut, Daniel Négru, Yiping Chen, Pantelis A. Frangoudis, and Adlen Ksentini. "Content delivery networks as a virtual network function: a win-win isp-cdn collaboration." *In Global Communications Conference (GLOBECOM), IEEE, pp. 1-6. IEEE, 2016.*

3. Nicolas Herbaut, Daniel Négru, Damien Magoni, and Pantelis A. Frangoudis. "Deploying a content delivery service function chain on an SDN-NFV operator infrastructure." *In Telecommunications and Multimedia (TEMU), International Conference on, pp. 1-7. IEEE, 2016.*

4. Nicolas Herbaut, Daniel Négru, George Xilouris, and Yiping Chen. "Migrating to a nfv-based home gateway: introducing a surrogate vnf approach." *In Network of the Future (NOF), 6th International Conference on the, pp. 1-7. IEEE, 2015.*

National Conference/Workshops

1. Nicolas Herbaut, François-Vivien Guiot, and Daniel Négru "Un Modèle De Collaboration Pour La Diffusion De Contenus Basé Sur L'utilisation De La Blockchain" *The Big Block Theory'17, Paris, 1ère Conférence Scientifique sur les Technologies de Registres Distribués, 04/2017.*

2. Nicolas Herbaut, and François-Vivien Guiot, "Fiabilité et sincérité des systèmes blockchain" *Colloque Convergence Droit et Numérique: Université de Bordeaux, 2017.*

3. Nicolas Herbaut, "La mutation des opérateurs réseaux en fournisseurs de Cloud: quels enjeux pour la neutralité du net? ", *Atelier Convergence Droit et Numérique de l'Université de Bordeaux, 2017.*

4. Nicolas Herbaut, David Bourasseau, and Maxime Peterlin, "Développer, Tester et Livrer des Microservices à l'aide de Docker", *JDEV2015 (Journées Nationales du développement Logiciel de L'Enseignement Supérieur et de la Recherche), Bordeaux, CNRS, 07/2015.*

Bibliography

[Abid et al., 2014] Abid, A., Khemakhem, M. T., Marzouk, S., Jemaa, M. B., Monteil, T., and Drira, K. (2014). Toward antifragile cloud computing infrastructures. *Procedia Computer Science*, 32:850–855. Cité page 45.

[Abujoda and Papadimitriou, 2015] Abujoda, A. and Papadimitriou, P. (2015). Midas: Middlebox discovery and selection for on-path flow processing. In *7th International Conference on Communication Systems and Networks*. Cité page 92.

[Achterberg, 2009] Achterberg, T. (2009). Scip: solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41. Cité page 107.

[Adhikari et al., 2012] Adhikari, V. K., Guo, Y., Hao, F., Varvello, M., Hilt, V., Steiner, M., and Zhang, Z.-L. (2012). Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *INFOCOM, 2012 Proceedings IEEE*, pages 1620–1628. IEEE. Cité pages 28 et 59.

[Aflatoonian et al., 2015] Aflatoonian, A., Bouabdallah, A., Guillouard, K., Catros, V., and Bonnin, J.-M. (2015). Byoc: Bring your own control a new concept to monetize sdn's openness. In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pages 1–5. IEEE. Cité pages 22 et 64.

[Akamai Technologies, 2008] Akamai Technologies (2008). Akamai's state of the internet. Technical Report Volume 1 Number 2, Akamai. Cité page 18.

[Akamai Technologies, 2017] Akamai Technologies (2017). Akamai's state of the internet. Technical Report Volume 9 Number 1, Akamai. Cité page 18.

[Alimi et al., 2014] Alimi, R., Yang, Y., and Penno, R. (2014). Application-layer traffic optimization (ALTO) protocol. RFC 7285, IEFT. Cité page 58.

[Amaldi et al., 2016] Amaldi, E., Coniglio, S., Koster, A. M., and Tieves, M. (2016). On the computational complexity of the virtual network embedding problem. *Electronic Notes in Discrete Mathematics*, 52:213–220. Cité page 102.

[Atzei et al., 2017] Atzei, N., Bartoletti, M., and Cimoli, T. (2017). A survey of attacks on ethereum smart contracts (sok). In *International Conference on Principles of Security and Trust*, pages 164–186. Springer. Cité page 123.

[AWS, 2017] AWS, A. (2017). Amazon ec2 instance types – amazon web services (aws). Cité page 63.

[Bari et al., 2015] Bari, M. F., Chowdhury, S. R., Ahmed, R., and Boutaba, R. (2015). On orchestrating virtual network functions in NFV. *CoRR*. Cité page 91.

[Belli, 2016] Belli, L. (2016). End-to-end, net neutrality and human rights. In *Net Neutrality Compendium*, pages 13–29. Springer. Cité page 73.

[Benson et al., 2011] Benson, T., Akella, A., Shaikh, A., and Sahu, S. (2011). Cloudnaas: A cloud networking platform for enterprise applications. In *Proceedings of the 2Nd ACM Symposium on Cloud Computing*, SOCC '11, pages 8:1–8:13, New York, NY, USA. ACM. Cité page 91.

[BEREC, 2016] BEREC (2016). Berec guidelines on the implementation by national regulators of european net neutrality rules. Technical Report BoR (16) 127, BEREC. http://berec.europa.eu/eng/document_register/subject_matter/berec/download/0/6160-berec-guidelines-on-the-implementation-b_0.pdf. Cité page 73.

[Bertin et al., 2017] Bertin, P., Mamouni, T., and Gosselin, S. (2017). Next-generation pop with functional convergence redistributions. In *Fiber-Wireless Convergence in Next-Generation Communication Networks*, pages 319–336. Springer. Cité page 29.

[Bertrand et al., 2012] Bertrand, G., Stephan, E., Burbridge, T., Eardley, P., Ma, K., and Watson, G. (2012). Use cases for content delivery network interconnection. Technical report, IETF. Cité page 28.

[Bhamare et al., 2017] Bhamare, D., Samaka, M., Erbad, A., Jain, R., Gupta, L., and Chan, H. A. (2017). Optimal virtual network function placement in multi-cloud service function chaining architecture. *Computer Communications*, 102:1–16. Cité page 92.

[Bolla et al., 2014] Bolla, R., Lombardo, C., Bruschi, R., and Mangialardi, S. (2014). Dropv2: energy efficiency through network function virtualization. *IEEE Network*, 28(2):26–32. Cité page 92.

[Bonola et al., 2017] Bonola, M., Bifulco, R., Petrucci, L., Pontarelli, S., Tulumello, A., and Bianchi, G. (2017). Implementing advanced network functions for datacenters with stateful programmable data planes. In *Local and Metropolitan Area Networks (LANMAN), 2017 IEEE International Symposium on*, pages 1–6. IEEE. Cité page 51.

[Bosshart et al., 2014] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., et al. (2014). P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, 44(3):87–95. Cité page 7.

[Böttger et al., 2016] Böttger, T., Cuadrado, F., Tyson, G., Castro, I., and Uhlig, S. (2016). Open connect everywhere: A glimpse at the internet ecosystem through the lens of the netflix cdn. *arXiv preprint arXiv:1606.05519*. Cité pages 1 et 76.

[Bouten et al., 2015] Bouten, N., Famaey, J., Mijumbi, R., Naudts, B., Serrat, J., Latré, S., and De Turck, F. (2015). Towards nfv-based multimedia delivery. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 738–741. IEEE. Cité pages 29 et 105.

[Bozic et al., 2016] Bozic, N., Pujolle, G., and Secci, S. (2016). A tutorial on blockchain and applications to secure network control-planes. In *Smart Cloud Networks & Systems (SCNS)*, pages 1–8. IEEE. Cité page 77.

[Broadband Forum, 2005] Broadband Forum (2005). Base requirements for an adsl modem with routing. TECHNICAL REPORT TR-068, Broadband Forum. Cité page 18.

[Broadband Forum, 2006] Broadband Forum (2006). Functional Requirements for Broadband Residential Gateway Devices. TECHNICAL REPORT TR-124, Broadband Forum. Cité page 18.

[Buterin, 2015] Buterin, V. (2015). On public and private blockchains - ethereum blog. https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/. Cité page 81.

[Callegati et al., 2017] Callegati, F., Cerroni, W., Contoli, C., and Foresta, F. (2017). Performance of intent-based virtualized network infrastructure management. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE. Cité page 90.

[Cantó Palancar et al., 2015] Cantó Palancar, R., López da Silva, R. A., Folgueira Chavarría, J. L., López, D. R., Elizondo Armengol, A. J., and Gamero Tinoco, R. (2015). Virtualization of residential customer premise equipment. lessons learned in brazil vcpe trial. *it-Information Technology*, 57(5):285–294. Cité pages 9 et 21.

[Castro et al., 1999] Castro, M., Liskov, B., et al. (1999). Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186. Cité page 77.

[Chellouche et al., 2012] Chellouche, S. A., Negru, D., Chen, Y., and Sidibe, M. (2012). Home-box-assisted content delivery network for internet video-on-demand services. In *Computers and Communications (ISCC), 2012 IEEE Symposium on*, pages 000544–000550. IEEE. Cité page 28.

[Cho et al., 2011] Cho, K., Jung, H., Lee, M., Ko, D., Kwon, T., and Choi, Y. (2011). How can an isp merge with a cdn? *IEEE Communications Magazine*, 49(10):156–162. Cité page 28.

[Chowdhury et al., 2012] Chowdhury, M., Rahman, M. R., and Boutaba, R. (2012). Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. *IEEE/ACM Transactions on Networking (TON)*, 20(1):206–219. Cité page 100.

[Chuang, 2011] Chuang, J. (2011). Loci of competition for future internet architectures. *IEEE Communications Magazine*, 49(7):38–43. Cité page 76.

[Cisco, 2016] Cisco, V. N. I. (2016). Forecast and methodology, 2015-2020. *White Paper, Cisco*. Cité page 26.

[Cohen et al., 2013] Cohen, R., Barabash, K., Rochwerger, B., Schour, L., Crisan, D., Birke, R., Minkenberg, C., Gusat, M., Recio, R., and Jain, V. (2013). An intent-based approach for network virtualization. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 42–50. IEEE. Cité page 90.

[Cohen et al., 2015] Cohen, R., Lewin-Eytan, L., Naor, J. S., and Raz, D. (2015). Near optimal placement of virtual network functions. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1346–1354. Cité page 91.

[Comi et al., 2016] Comi, P., Crosta, P. S., Beccari, M., Paglierani, P., Grossi, G., Pedersini, F., and Petrini, A. (2016). Hardware-accelerated high-resolution video coding in virtual network functions. In *Networks and Communications (EuCNC), 2016 European Conference on*, pages 32–36. IEEE. Cité page 54.

[Council et al., 2001] Council, N. R. et al. (2001). *Looking over the Fence at Networks: A Neighbor's View of Networking Research*. National academies Press. https://www.nap.edu/download/10183. Cité page 8.

[Croman et al., 2016] Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G., et al. (2016). On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125. Springer. Cité page 87.

[Cziva and Pezaros, 2017] Cziva, R. and Pezaros, D. P. (2017). Container network functions: Bringing nfv to the network edge. *IEEE Communications Magazine*, 55(6):24–31. Cité page 83.

[da Silva Gonçalves, 2016] da Silva Gonçalves, V. G. (2016). Online video in the future internet age: business and policy dynamics. *PhD dissertation*. Cité page 75.

[David McLaughlin, 2015] David McLaughlin, T. S. (2015). U.S. Antitrust Lawyers Said Leaning Against Comcast Deal. *Bloomberg Technology*. https://www.bloomberg.com/news/articles/2015-04-17/ u-s-antitrust-lawyers-said-to-be-leaning-against-comcast-merger. Cité page 28.

[David Minodier, and Gregory Dalle, Juniper, 2016] David Minodier, and Gregory Dalle, Juniper (2016). Network Enhanced Residential Gateway. Technical Report TR - 317, Broadband Forum. https://www.broadband-forum.org/technical/download/TR-317.pdf. Cité page 82.

[Den Hartog et al., 2004] Den Hartog, F., Balm, M., De Jong, C., and Kwaaitaai, J. (2004). Convergence of residential gateway technology. *IEEE Communications Magazine*, 42(5):138–143. Cité page 18.

[Dietrich et al., 2015] Dietrich, D., Abujoda, A., and Papadimitriou, P. (2015). Network service embedding across multiple providers with nestor. In *IFIP Networking Conference (IFIP Networking), 2015*. Cité page 91.

[Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271. Cité page 103.

[Dinh et al., 2017] Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., and Tan, K.-L. (2017). Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pages 1085–1100, New York, NY, USA. ACM. Cité page 87.

[Dobrian et al., 2011] Dobrian, F., Sekar, V., Awan, A., Stoica, I., Joseph, D., Ganjam, A., Zhan, J., and Zhang, H. (2011). Understanding the impact of video quality on user engagement. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 362–373. ACM. Cité page 58.

[Dong et al., 2012] Dong, Y., Yang, X., Li, J., Liao, G., Tian, K., and Guan, H. (2012). High performance network virtualization with sr-iov. *Journal of Parallel and Distributed Computing*, 72(11):1471–1480. Cité page 9.

[ETSI, 2013] ETSI (2013). Network Functions Virtualisation (NFV); Use Cases. Technical Report ETSI GS NFV 001 V1.1.1, ETSI. Cité page 35.

[ETSI, 2013a] ETSI, G. (2013a). Network functions virtualisation (nfv): Architectural framework. *ETsI Gs NFV*, 2(2):V1. Cité pages 10 et 35.

[ETSI, 2013b] ETSI, I. (2013b). Gs-nfv-003 network functions virtualisation (nfv). *Terminlogy for main concepts in NFV*, 10. Cité page 49.

[ETSI, 2015] ETSI, I. (2015). Etsi gs nfv-rel 001 v1. 1.1: Network functions virtualisation(nfv); resiliency requirements. Cité page 25.

[ETSI, 2012] ETSI, N. (2012). Network functions virtualization, white paper. Cité page 8.

[ETSI, Network Functions VirtualisationV., 2014] ETSI, Network Functions VirtualisationV. (2014). Management and orchestration. *ETSI GS NFV-MAN*, 1:V1. Cité pages 13, 46 et 47.

[ETSI TISPAN, 2011] ETSI TISPAN (2011). Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Content Delivery Network (CDN) Architecture. *ETSI Standards*, V3.1.2(V3.1.2). http://www.etsi.org/deliver/etsi_ts/182000_182099/182019/03.01.02_60/ts_182019v030102p.pdf. Cité page 30.

[European Commission, 2016] European Commission (2016). Broadband Internet access cost (BIAC) report Autumn 2015 - EU Law and Publications. https://publications.europa.eu/en/publication-detail/-/publication/d7952782-9aea-11e6-868c-01aa75ed71a1. Cité page 20.

[Evans, 2009] Evans, D. S. (2009). The online advertising industry: Economics, evolution, and privacy. *The journal of economic perspectives*, 23(3):37–60. Cité page 18.

[Famaey et al., 2012a] Famaey, J., Latre, S., Wauters, T., and De Turck, F. (2012a). An SLA-driven framework for dynamic multimedia content delivery federations. In *Proc. IEEE NOMS*. Cité page 63.

[Famaey et al., 2012b] Famaey, J., Latré, S., Wauters, T., and De Turck, F. (2012b). An sla-driven framework for dynamic multimedia content delivery federations. In *2012 IEEE Network Operations and Management Symposium*, pages 1241–1247. IEEE. Cité page 114.

[Feamster et al., 2007] Feamster, N., Gao, L., and Rexford, J. (2007). How to lease the internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 37(1):61–64. Cité pages 20 et 22.

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

[Felter et al., 2015] Felter, W., Ferreira, A., Rajamony, R., and Rubio, J. (2015). An updated performance comparison of virtual machines and linux containers. In *Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium On*, pages 171–172. IEEE. Cité page 44.

[Ferguson, 2006] Ferguson, D. (2006). P2p file sharing–the evolving distribution chain. *CacheLogic, WDC.* Cité page 19.

[Ficsor, 2001] Ficsor, M. (2001). *Law of Copyright and the Internet: The Wipo Treaties and Their Implementation.* Oxford University Press. Cité page 19.

[Fischer et al., 2013] Fischer, A., Botero, J., Till Beck, M., de Meer, H., and Hesselbach, X. (2013). Virtual Network Embedding: A Survey. *IEEE Communications Surveys Tutorials*, 15(4). Cité page 100.

[Foundation, 2016] Foundation, O. N. (2016). Intent NBI – Definition and Principles. Technical Recommandation TR-523, Open Networking Foundation. Cité page 91.

[Fowler and Lewis, 2014] Fowler, M. and Lewis, J. (2014). Microservices. *ThoughtWorks. http://martinfowler.com/articles/microservices.html [last accessed on February 17, 2015].* Cité page 82.

[Frangoudis et al., 2017] Frangoudis, P. A., Yala, L., and Ksentini, A. (2017). Cdn-as-a-service provision over a telecom operator's cloud. *IEEE Transactions on Network and Service Management.* Cité page 35.

[Frangoudis et al., 2016] Frangoudis, P. A., Yala, L., Ksentini, A., and Taleb, T. (2016). An architecture for on-demand service deployment over a telco cdn. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE. Cité page 35.

[Frank et al., 2013] Frank, B., Poese, I., Lin, Y., Smaragdakis, G., Feldmann, A., Maggs, B., Rake, J., Uhlig, S., and Weber, R. (2013). Pushing cdn-isp collaboration to the limit. *ACM SIGCOMM Computer Communication Review*, 43(3):34–44. Cité page 58.

[Freeman and Boutaba, 2016] Freeman, H. and Boutaba, R. (2016). Networking industry transformation through softwarization. *IEEE Communications Magazine*, 54(8):4–6. Cité page 5.

[Garay et al., 2016] Garay, J., Matias, J., Unzilla, J., and Jacob, E. (2016). Service description in the nfv revolution: Trends, challenges and a way forward. *IEEE Communications Magazine*, 54(3):68–74. Cité page 90.

[Gardikis et al., 2016] Gardikis, G., Koutras, I., Mavroudis, G., Costicoglou, S., Xilouris, G., Sakkas, C., and Kourtis, A. (2016). An integrating framework for efficient nfv monitoring. In *NetSoft Conference and Workshops (NetSoft), 2016 IEEE*, pages 1–5. IEEE. Cité page 15.

[Gember et al., 2013] Gember, A., Krishnamurthy, A., St. John, S., Grandl, R., Gao, X., Anand, A., Benson, T., Sekar, V., and Akella, A. (2013). Stratos: A Network-Aware Orchestration Layer for Virtual Middleboxes in Clouds. *ArXiv e-prints.* Cité page 91.

[Gharakheili et al., 2016] Gharakheili, H. H., Vishwanath, A., and Sivaraman, V. (2016). An economic model for a new broadband ecosystem based on fast and slow lanes. *IEEE Network*, 30(2):26–31. Cité page 60.

[Giotis et al., 2015] Giotis, K., Kryftis, Y., and Maglaris, V. (2015). Policy-based orchestration of nfv services in software-defined networks. In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pages 1–5. IEEE. Cité page 35.

[Hallingby et al., 2016] Hallingby, H. K., Hartviksen, G., Elaluf-Calderwood, S., and Sørensen, C. (2016). Convergence in action: A case study of the norwegian internet. *Telematics and Informatics*, 33(2):641–649. Cité page 1.

[Han et al., 2015] Han, B., Gopalakrishnan, V., Ji, L., and Lee, S. (2015). Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2):90–97. Cité page 21.

[Heller et al., 2012] Heller, B., Sherwood, R., and McKeown, N. (2012). The controller placement problem. In *Proceedings of the first workshop on Hot topics in software defined networks*, pages 7–12. ACM. Cité page 96.

[Herbaut and Négru, ress] Herbaut, N. and Négru, D. (In Press). A model for collaborative blockchain-based video delivery relying on advanced network services chains. *IEEE Communication Magazine*, 55. Cité page 3.

[Herbaut et al., 2016] Herbaut, N., Négru, D., Chen, Y., Frangoudis, P. A., and Ksentini, A. (2016). Content delivery networks as a virtual network function: a win-win isp-cdn collaboration. In *IEEE Global Communications Conference (GLOBECOM)*, Washington DC. IEEE, IEEE. Cité page 3.

[Herbaut et al., 2017a] Herbaut, N., Négru, D., Dietrich, D., and Papadimitriou, P. (2017a). Dynamic deployment and optimization of virtual content delivery networks. *IEEE Multimedia*, 24. Cité page 3.

[Herbaut et al., 2017b] Herbaut, N., Négru, D., Dietrich, D., and Papadimitriou, P. (2017b). Service chain modeling and embedding for nfv-based content delivery. In *IEEE International Conference on Communications (ICC)*. Cité page 3.

[Herbaut et al., 2015] Herbaut, N., Négru, D., Xilouris, G., and Chen, Y. (2015). Migrating to a nfv-based home gateway: introducing a surrogate vnf approach. In *Network of the Future (NOF), 2015 6th International Conference on the*, Montréal. IEEE, IEEE. Cité page 81.

[Hernandez-Valencia et al., 2015] Hernandez-Valencia, E., Izzo, S., and Polonsky, B. (2015). How will nfv/sdn transform service provider opex? *IEEE Network*, 29(3):60–67. Cité page 21.

[Herrera and Botero, 2016] Herrera, J. G. and Botero, J. F. (2016). Resource allocation in nfv: A comprehensive survey. *IEEE Transactions on Network and Service Management*, 13(3):518–532. Cité page 90.

[Hippel and Krogh, 2003] Hippel, E. v. and Krogh, G. v. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. *Organization science*, 14(2):209–223. Cité page 8.

[Holliday, 1997] Holliday, C. R. (1997). The residential gateway. *IEEE spectrum*, 34(5):29–31. Cité page 18.

[Huang et al., 2017] Huang, W., Zhu, H., and Qian, Z. (2017). Autovnf: An automatic resource sharing schema for vnf requests. *Journal of Internet Services and Information Security (JISIS)*, 7(3):34–47. Cité page 92.

[Hugenholtz et al., 2000] Hugenholtz, B. et al. (2000). Why the copyright directive is unimportant, and possibly invalid. *European Intellectual Property Review*, 22(11):499–505. Cité page 19.

[Hull et al., 2016] Hull, R., Batra, V. S., Chen, Y.-M., Deutsch, A., Heath III, F. F. T., and Vianu, V. (2016). Towards a shared ledger business collaboration language based on data-aware processes. In *International Conference on Service-Oriented Computing*, pages 18–36. Springer. Cité page 79.

[Hunt et al., 2010] Hunt, P., Konar, M., Junqueira, F. P., and Reed, B. (2010). Zookeeper: Wait-free coordination for internet-scale systems. In *USENIX annual technical conference*, volume 8, page 9. Boston, MA, USA. Cité page 40.

[Hyndman and Koehler, 2006] Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688. Cité page 113.

[IETF, 2001] IETF (2001). Rfc 3040: Internet web replication and caching taxonomy. Technical report, RFC 3040. Cité page 30.

[Intel, 2015] Intel, D. (2015). Data plane development kit. Cité page 51.

[Jain et al., 2013] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., et al. (2013). B4: Experience with a globally-deployed software defined wan. *ACM SIGCOMM Computer Communication Review*, 43(4):3–14. Cité page 7.

[Jentzsch, 2016] Jentzsch, C. (2016). Decentralized autonomous organization to automate governance. *Online-Publikation: https://download.slock.it/public/DAO/WhitePaper.pdf*. Cité page 81.

[Jiang et al., 2016] Jiang, H., Bouabdallah, A., Aflatoonian, A., Bonnin, J.-M., and Guillouard, K. (2016). A secure multi-tenant framework for sdn. In *Proceedings of the 9th International Conference on Security of Information and Networks*, pages 40–44. ACM. Cité page 64.

[Jiang et al., 2009] Jiang, W., Zhang-Shen, R., Rexford, J., and Chiang, M. (2009). Cooperative content distribution and traffic engineering in an ISP network. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '09, pages 239–250. ACM. Cité page 58.

[Karras et al., 2016] Karras, K., Blott, M., and Vissers, K. A. (2016). Virtualization of programmable integrated circuits. US Patent 9,503,093. Cité page 9.

[Kaushal and Bala, 2011] Kaushal, V. and Bala, A. (2011). Autonomic fault tolerance using haproxy in cloud environment. *Int. J. of Advanced Engineering Sciences and Technologies*, 7(2):54–59. Cité page 40.

[Kelly et al., 2012] Kelly, T. J. C., Rossotto, C. M., et al. (2012). *Broadband strategies handbook*. World Bank Publications. Cité page 18.

[Kim and Lee, 2014] Kim, T. and Lee, B. (2014). Scalable cdn service poc over distributed cloud management platform. In *Information and Communication Technology Convergence (ICTC), 2014 International Conference on*, pages 832–833. IEEE. Cité page 35.

[Knight et al., 2011] Knight, S., Nguyen, H. X., Falkner, N., Bowden, R., and Roughan, M. (2011). The internet topology zoo. *Selected Areas in Communications, IEEE Journal on*, 29(9). Cité page 106.

[Kourtis et al., 2017] Kourtis, M.-A., McGrath, M. J., Gardikis, G., Xilouris, G., Riccobene, V., Papadimitriou, P., Trouva, E., Liberati, F., Trubian, M., Batallé, J., et al. (2017). T-nova: An open-source mano stack for nfv infrastructures. *IEEE Transactions on Network and Service Management*. Cité page 10.

[Kourtis et al., 2015] Kourtis, M.-A., Xilouris, G., Riccobene, V., McGrath, M. J., Petralia, G., Koumaras, H., Gardikis, G., and Liberal, F. (2015). Enhancing vnf performance by exploiting sr-iov and dpdk packet processing acceleration. In *Network Function Virtualization and Software Defined Network (NFV-SDN), 2015 IEEE Conference on*, pages 74–78. IEEE. Cité page 51.

[Kreitz and Niemela, 2010] Kreitz, G. and Niemela, F. (2010). Spotify–large scale, low latency, p2p music-on-demand streaming. In *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on*, pages 1–10. IEEE. Cité page 27.

[Kreutz et al., 2015] Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76. Cité page 6.

[Krishnan et al., 2009a] Krishnan, R., Madhyastha, H. V., Srinivasan, S., Jain, S., Krishnamurthy, A., Anderson, T., and Gao, J. (2009a). Moving beyond end-to-end path information to optimize cdn performance. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 190–201. ACM. Cité page 34.

[Krishnan et al., 2009b] Krishnan, R., Madhyastha, H. V., Srinivasan, S., Jain, S., Krishnamurthy, A., Anderson, T., and Gao, J. (2009b). Moving beyond end-to-end path information to optimize cdn performance. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 190–201. ACM. Cité page 58.

[Kryftis et al., 2016] Kryftis, Y., Mastorakis, G., Mavromoustakis, C. X., Batalla, J. M., Pallis, E., and Kormentzas, G. (2016). Efficient entertainment services provision over a novel network architecture. *IEEE Wireless Communications*, 23(1):14–21. Cité pages 54, 80 et 113.

[Lee et al., 2013] Lee, T., Chen, T., and Milano, E. D. T. (2013). Library and resources for third party apps for smarttv. US Patent App. 13/969,777. Cité page 20.

[Levy, 2017] Levy, K. E. (2017). Book-smart, not street-smart: Blockchain-based smart contracts and the social workings of law. *Engaging Science, Technology, and Society*, 3:1–15. Cité page 123.

[Li et al., 2017] Li, W., Sforzin, A., Fedorov, S., and Karame, G. O. (2017). Towards scalable and private industrial blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pages 9–14. ACM. Cité page 87.

[Liberati et al., 2017] Liberati, F., Giuseppi, A., Pietrabissa, A., Suraci, V., Di Giorgio, A., Trubian, M., Dietrich, D., Papadimitriou, P., and Delli Priscoli, F. (2017). Stochastic and exact methods for service mapping in virtualized network infrastructures. *International Journal of Network Management*. Cité page 14.

[Lukovszki et al., 2016] Lukovszki, T., Rost, M., and Schmid, S. (2016). It's a match!: Near-optimal and incremental middlebox deployment. *SIGCOMM Comput. Commun. Rev.*, 46(1):30–36. Cité page 91.

[Lukovszki and Schmid, 2014] Lukovszki, T. and Schmid, S. (2014). Online admission control and embedding of service chains. In *Structural Information and Communication Complexity*, pages 104–118. Springer. Cité page 91.

[Lunney Jr, 2001] Lunney Jr, G. S. (2001). The death of copyright: Digital technology, private copying, and the digital millennium copyright act. *Virginia Law Review*, pages 813–920. Cité page 19.

[Luo et al., 2009] Luo, J.-G., Zhang, Q., Tang, Y., and Yang, S.-Q. (2009). A trace-driven approach to evaluate the scalability of p2p-based video-on-demand service. *IEEE Transactions on Parallel and Distributed Systems*, 20(1):59–70. Cité page 28.

[Manthena et al., 2015] Manthena, M. P. V., van Adrichem, N. L., van den Broek, C., and Kuipers, F. (2015). An sdn-based architecture for network-as-a-service. In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pages 1–5. IEEE. Cité page 22.

[Markakis et al., 2016] Markakis, E., Sideris, A., Alexiou, G., Bourdena, A., Pallis, E., Mastorakis, G., and Mavromoustakis, C. X. (2016). A virtual network functions brokering mechanism. In *Telecommunications and Multimedia (TEMU), 2016 International Conference on*, pages 1–5. IEEE. Cité page 15.

[Mathis et al., 1997] Mathis, M., Semke, J., Mahdavi, J., and Ott, T. (1997). The macroscopic behavior of the tcp congestion avoidance algorithm. *ACM SIGCOMM Computer Communication Review*, 27(3):67–82. Cité page 93.

[Matloff, 2008] Matloff, N. (2008). Introduction to discrete-event simulation and the simpy language. *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August*, 2:2009. Cité page 83.

[McKeown et al., 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74. Cité page 6.

[Mehraghdam et al., 2014a] Mehraghdam, S., Keller, M., and Karl, H. (2014a). Specifying and placing chains of virtual network functions. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, pages 7–13. IEEE. Cité page 91.

[Mehraghdam et al., 2014b] Mehraghdam, S., Keller, M., and Karl, H. (2014b). Specifying and placing chains of virtual network functions. In *CloudNet, 2014 IEEE 3rd International Conference on*, pages 7–13. Cité page 91.

[Mijumbi et al., 2016] Mijumbi, R., Serrat, J., Gorricho, J.-L., Bouten, N., De Turck, F., and Boutaba, R. (2016). Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys & Tutorials*, 18(1):236–262. Cité page 7.

[Mijumbi et al., 2015] Mijumbi, R., Serrat, J., Gorricho, J.-L., Bouten, N., De Turck, F., and Davy, S. (2015). Design and evaluation of algorithms for mapping and scheduling of virtual network functions. In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pages 1–9. IEEE. Cité page 92.

[Miller et al., 2014] Miller, A., Juels, A., Shi, E., Parno, B., and Katz, J. (2014). Permacoin: Repurposing bitcoin work for data preservation. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 475–490. IEEE. Cité page 80.

[Muñoz et al., 2015] Muñoz, R., Vilalta, R., Casellas, R., Martínez, R., Szyrkowiec, T., Autenrieth, A., López, V., and López, D. (2015). Sdn/nfv orchestration for dynamic deployment of virtual sdn controllers as vnf for multi-tenant optical networks. In *Optical Fiber Communications Conference and Exhibition (OFC), 2015*, pages 1–3. IEEE. Cité page 22.

[Nakamoto, 2008] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Cité page 77.

[Naudts et al., 2017] Naudts, B., Flores, M., Mijumbi, R., Verbrugge, S., Serrat, J., and Colle, D. (2017). A dynamic pricing algorithm for a network of virtual resources. *International Journal of Network Management*, 27(2). Cité page 92.

[Nygren et al., 2010] Nygren, E., Sitaraman, R. K., and Sun, J. (2010). The akamai network: a platform for high-performance internet applications. *ACM SIGOPS Operating Systems Review*, 44(3):2–19. Cité pages 31 et 58.

[of Chairman Pai, 2017] of Chairman Pai, F. O. (2017). Factsheet - restoring internet freedom for all americans. Technical report, FCC. https://apps.fcc.gov/edocs_public/attachmatch/DOC-344592A1.pdf. Cité page 74.

[Paglierani, 2015] Paglierani, P. (2015). High performance computing and network function virtualization: A major challenge towards network programmability. In *Communications and Networking (BlackSeaCom), 2015 IEEE International Black Sea Conference on*, pages 137–141. IEEE. Cité pages 9 et 53.

[Pahl and Lee, 2015] Pahl, C. and Lee, B. (2015). Containers and clusters for edge cloud architectures–a technology review. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pages 379–386. IEEE. Cité page 83.

[Pallis and Vakali, 2006] Pallis, G. and Vakali, A. (2006). Insight and perspectives for content delivery networks. *Communications of the ACM*, 49(1):101–106. Cité page 54.

[Pathan, 2014a] Pathan, M. (2014a). Cloud-based content delivery and streaming. *Advanced Content Delivery, Streaming, and Cloud Services*, pages 1–31. Cité page 1.

[Pathan, 2014b] Pathan, M. (2014b). Cloud-based content delivery and streaming. In Pathan, M., Sitaraman, R. K., and Robinson, D., editors, *Advanced Content Delivery, Streaming, and Cloud Services*. Wiley. Cité page 34.

[Pathan et al., 2014] Pathan, M., Sitaraman, R. K., and Robinson, D. (2014). *Advanced content delivery, streaming, and cloud services*. John Wiley & Sons. Cité page 26.

[Peterson et al., 2014] Peterson, L., Davie, B., and Brandenburg, R. v. (2014). Framework for content distribution network interconnection (CDNI). RFC 7336, IETF. Cité page 58.

[Pfaff et al., 2015] Pfaff, B., Pettit, J., Koponen, T., Jackson, E. J., Zhou, A., Rajahalme, J., Gross, J., Wang, A., Stringer, J., Shelar, P., et al. (2015). The design and implementation of open vswitch. In *NSDI*, pages 117–130. Cité page 6.

[Pontarelli et al., 2017] Pontarelli, S., Bonola, M., and Bianchi, G. (2017). Smashing sdn" built-in" actions: Programmable data plane packet manipulation in hardware. In *Network Softwarization (NetSoft), 2017 IEEE Conference on*, pages 1–9. IEEE. Cité page 7.

[Poularakis et al., 2017] Poularakis, K., Iosifidis, G., Smaragdakis, G., and Tassiulas, L. (2017). One step at a time: Optimizing sdn upgrades in isp networks. In *Proceedings of IEEE INFOCOM*. Cité page 22.

[Proença et al., 2017] Proença, J., Cruz, T., Simões, P., Gaspar, G., Parreira, B., Laranjeira, A., and Bastos, F. (2017). Building an nfv-based vrgw: Lessons learned. In *Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual*, pages 653–658. IEEE. Cité page 9.

[Qu et al., 2016] Qu, L., Assi, C., and Shaban, K. (2016). Delay-aware scheduling and resource optimization with network function virtualization. *IEEE Transactions on Communications*, 64(9):3746–3758. Cité page 105.

[Quinn and Nadeau, 2015] Quinn, P. and Nadeau, T. (2015). Problem statement for service function chaining, rfc 7498. Cité page 100.

[Ramakrishnan et al., 2001] Ramakrishnan, K., Floyd, S., and Black, D. (2001). Rfc 3168 the addition of explicit congestion notification (ecn) to ip. Technical report, IETF. Cité page 58.

[Rebahi et al., 2016] Rebahi, Y., Ghamsi, M. S., Herbaut, N., Négru, D., Comi, P. M., Crosta, P. S., Lorenz, P., Pallis, E., and Markakis, E. (2016). Virtual network functions deployment between business expectations and technical challenges: The t-nova approach. *Recent Advances in Communications and Networking Technology*, 5. Cité page 3.

[Riera et al., 2016] Riera, J. F., Batallé, J., Bonnet, J., Días, M., McGrath, M., Petralia, G., Liberati, F., Giuseppi, A., Pietrabissa, A., Ceselli, A., et al. (2016). Tenor: Steps towards an orchestration platform for multi-pop nfv deployment. In *NetSoft Conference and Workshops (NetSoft), 2016 IEEE*, pages 243–250. IEEE. Cité page 10.

[Riera et al., 2014] Riera, J. F., Escalona, E., Batalle, J., Grasa, E., and Garcia-Espin, J. A. (2014). Virtual network function scheduling: Concept and challenges. In *Smart Communications in Network Technologies (SaCoNeT), 2014 International Conference on*, pages 1–5. IEEE. Cité page 92.

[Rimal et al., 2009] Rimal, B. P., Choi, E., and Lumb, I. (2009). A taxonomy and survey of cloud computing systems. *NCM*, 9:44–51. Cité page 94.

[Rodriguez-Natal et al., 2017] Rodriguez-Natal, A., Ermagan, V., Noy, A., Sahai, A., Kaempfer, G., Barkai, S., Maino, F., and Cabellos-Aparicio, A. (2017). Global state, local decisions: Decentralized nfv for isps via enhanced sdn. *IEEE Communications Magazine*, 55(4):87–93. Cité page 22.

[Saba, 2009] Saba, J. (2009). Specifics on newspapers from'state of news media'report. *Editor & Publisher*. Cité page 18.

[Sahhaf et al., 2015] Sahhaf, S., Tavernier, W., Rost, M., Schmid, S., Colle, D., Pickavet, M., and Demeester, P. (2015). Network service chaining with optimized network function embedding supporting service decompositions. *Comput. Netw.*, 93:492–505. Cité page 91.

[SANDVINE, 2016] SANDVINE, I. (2016). Global internet phenomena report, latin america & north america. Cité pages 1 et 19.

[Sang and Li, 2002] Sang, A. and Li, S.-q. (2002). A predictability analysis of network traffic. *Computer networks*, 39(4):329–345. Cité page 113.

[Schechter, 2016] Schechter, E. (2016). Google online security blog: Adding youtube and calendar to the https transparency report. Cité page 57.

[Seaman et al., 2012] Seaman, C., Guo, Y., Izurieta, C., Cai, Y., Zazworka, N., Shull, F., and Vetrò, A. (2012). Using technical debt data in decision making: Potential decision approaches. In *Proceedings of the Third International Workshop on Managing Technical Debt*, pages 45–48. IEEE Press. Cité page 118.

[Seedorf et al., 2015] Seedorf, J., Yang, Y., and Peterson, J. (2015). CDNI Footprint and Capabilities Advertisement using ALTO. Internet-Draft draft-seedorf-cdni-request-routing-alto-08, IETF. Work in Progress. Cité page 58.

[Sengupta et al., 2016] Sengupta, B., Bag, S., Ruj, S., and Sakurai, K. (2016). Retricoin: Bitcoin based on compact proofs of retrievability. In *Proceedings of the 17th International Conference on Distributed Computing and Networking*, page 14. ACM. Cité page 80.

[Shalimov et al., 2013] Shalimov, A., Zuikov, D., Zimarina, D., Pashkov, V., and Smeliansky, R. (2013). Advanced study of sdn/openflow controllers. In *Proceedings of the 9th central & eastern european software engineering conference in russia*, page 1. ACM. Cité page 40.

[Shariatmadari et al., 2015] Shariatmadari, H., Ratasuk, R., Iraji, S., Laya, A., Taleb, T., Jäntti, R., and Ghosh, A. (2015). Machine-type communications: current status and future

perspectives toward 5g systems. *IEEE Communications Magazine*, 53(9):10–17. Cité page 29.

[Sherry et al., 2012] Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S., and Sekar, V. (2012). Making middleboxes someone else's problem: network processing as a cloud service. *ACM SIGCOMM Computer Communication Review*, 42(4):13–24. Cité page 8.

[Silva et al., 2013] Silva, G. C., Rose, L. M., and Calinescu, R. (2013). A systematic review of cloud lock-in solutions. In *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, volume 2, pages 363–368. IEEE. Cité pages 25 et 29.

[Singh et al., 2015] Singh, A., Ong, J., Agarwal, A., Anderson, G., Armistead, A., Bannon, R., Boving, S., Desai, G., Felderman, B., Germano, P., et al. (2015). Jupiter rising: A decade of clos topologies and centralized control in google's datacenter network. *ACM SIGCOMM Computer Communication Review*, 45(4):183–197. Cité pages 7 et 22.

[Skoviera et al., 2017] Skoviera, M., Harsh, P., Serhiienko, O., Belmonte, M. P., and Bohnert, T. M. (2017). Monetization of infrastructures and services. In *Networks and Communications (EuCNC), 2017 European Conference on*, pages 1–5. IEEE. Cité page 15.

[SONATA Consortium, 2015] SONATA Consortium (2015). D2.2 architecture design. Cité page 45.

[Srinivas and Patnaik, 1994] Srinivas, M. and Patnaik, L. M. (1994). Genetic algorithms: A survey. *computer*, 27(6):17–26. Cité page 104.

[Suh et al., 2014] Suh, M., Park, S. H., Lee, B., and Yang, S. (2014). Building firewall over the software-defined network controller. In *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, pages 744–748. IEEE. Cité page 7.

[Szabo, 1997] Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, 2(9). Cité page 78.

[T-NOVA Consortium, 2015] T-NOVA Consortium (2015). D2.22 overall system architecture and interfaces. Cité page 46.

[T-NOVA Consortium, 2016a] T-NOVA Consortium (2016a). D2.32 - specification of the infrastructure virtualisation, management and orchestration. Cité page 11.

[T-NOVA Consortium, 2016b] T-NOVA Consortium (2016b). Deliverable 5.32- network functions implementation and testing. http://www.t-nova.eu/results/. Cité pages 45 et 106.

[T-NOVA Consortium, 2017] T-NOVA Consortium (2017). D7.2 integrated pilot and validation report. Cité pages 45 et 48.

[Tang et al., 2017] Tang, M., Gao, L., Pang, H., Huang, J., and Sun, L. (2017). Optimizations and economics of crowdsourced mobile streaming. *IEEE Communications Magazine*, 55(4):21–27. Cité page 122.

[Trajkovska et al., 2017] Trajkovska, I., Kourtis, M.-A., Sakkas, C., Baudinot, D., Silva, J., Harsh, P., Xylouris, G., Bohnert, T. M., and Koumaras, H. (2017). Sdn-based service function chaining mechanism and service prototype implementation in nfv scenario. *Computer Standards & Interfaces*, 54:247–265. Cité page 50.

[Tuncer et al., 2013] Tuncer, D., Charalambides, M., Landa, R., and Pavlou, G. (2013). More control over network resources: An isp caching perspective. In *Network and Service Management (CNSM), 2013 9th International Conference on*, pages 26–33. IEEE. Cité page 27.

[van Brandenburg et al., 2015] van Brandenburg, R., Peterson, L., and Davie, B. (2015). Framework for Content Distribution Network Interconnection (CDNI). RFC 7336. Cité pages 30 et 58.

[Viginier, 2017] Viginier, P. (2017). Telco Transformation - Viginier: OrangeTransforms With DevOps. https://goo.gl/YhvQd4. Cité page 21.

[Virtualisation, 2012] Virtualisation, N. F. (2012). Introductory white paper. In *SDN and OpenFlow World Congress, Darmstadt, Germany*. Cité page 2.

[Vukolić, 2015] Vukolić, M. (2015). The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International Workshop on Open Problems in Network Security*, pages 112–125. Springer. Cité page 85.

[Wang et al., 2014] Wang, Y., Hu, Q., and Cao, X. (2014). Connectivity as a service: Towards optical-based network virtualization. In *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pages 264–268. IEEE. Cité page 62.

[Westkamp, 2007] Westkamp, G. (2007). The implementation of directive 2001/29/ec in the member states. *Queen Mary Intellectual Property Research Institute, London*. Cité page 19.

[Xilouris et al., 2015] Xilouris, G., Kourtis, M.-A., McGrath, M. J., Riccobene, V., Petralia, G., Markakis, E., Palis, E., Georgios, A., Gardikis, G., Riera, J. F., et al. (2015). T-nova: Network functions as-a-service over virtualised infrastructures. In *Network Function Virtualization and Software Defined Network (NFV-SDN), 2015 IEEE Conference on*, pages 13–14. IEEE. Cité page 9.

[Xilouris et al., 2014] Xilouris, G., Trouva, E., Lobillo, F., Soares, J., Carapinha, J., McGrath, M. J., Gardikis, G., Paglierani, P., Pallis, E., Zuccaro, L., et al. (2014). T-nova: A marketplace

for virtualized network functions. In *Networks and Communications (EuCNC), 2014 European Conference on*, pages 1–5. IEEE. Cité page 14.

[Xu et al., 2016] Xu, X., Pautasso, C., Zhu, L., Gramoli, V., Ponomarev, A., Tran, A. B., and Chen, S. (2016). The blockchain as a software connector. In *Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference on*, pages 182–191. IEEE. Cité page 77.

[Xu et al., 2017] Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., Pautasso, C., and Rimba, P. (2017). A taxonomy of blockchain-based systems for architecture design. In *Software Architecture (ICSA), 2017 IEEE International Conference on*, pages 243–252. IEEE. Cité page 77.

[Yala et al., 2016] Yala, L., Frangoudis, P. A., and Ksentini, A. (2016). Qoe-aware computing resource allocation for cdn-as-a-service provision. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–6. IEEE. Cité page 35.

[Yap et al., 2017] Yap, K.-K., Motiwala, M., Rahe, J., Padgett, S., Holliman, M., Baldus, G., Hines, M., Kim, T., Narayanan, A., Jain, A., et al. (2017). Taking the edge off with espresso: Scale, reliability and programmability for global internet peering. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 432–445. ACM. Cité page 7.

[Yeganeh et al., 2013] Yeganeh, S. H., Tootoonchian, A., and Ganjali, Y. (2013). On scalability of software-defined networking. *IEEE Communications Magazine*, 51(2):136–141. Cité page 7.

[Yfoulis and Gounaris, 2009] Yfoulis, C. A. and Gounaris, A. (2009). Honoring slas on cloud computing services: a control perspective. In *Control Conference (ECC), 2009 European*, pages 184–189. IEEE. Cité page 94.

[Yin et al., 2009] Yin, H., Liu, X., Zhan, T., Sekar, V., Qiu, F., Lin, C., Zhang, H., and Li, B. (2009). Design and deployment of a hybrid cdn-p2p system for live video streaming: experiences with livesky. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 25–34. ACM. Cité page 28.

[Zhang et al., 2015] Zhang, G., Liu, W., Hei, X., and Cheng, W. (2015). Unreeling xunlei kankan: understanding hybrid cdn-p2p video-on-demand streaming. *IEEE Transactions on Multimedia*, 17(2):229–242. Cité page 28.

[Zhang et al., 2016] Zhang, W., Hwang, J., Rajagopalan, S., Ramakrishnan, K., and Wood, T. (2016). Flurries: Countless fine-grained nfs for flexible per-flow customization. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, pages 3–17. ACM. Cité page 92.

[Zheng et al., 2010] Zheng, G., Meneses, E., Bhatele, A., and Kale, L. V. (2010). Hierarchical load balancing for charm++ applications on large supercomputers. In *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on*, pages 436–444. IEEE. Cité page 52.

[Zouarhi, 2017] Zouarhi, S. (2017). Kidner–a worldwide decentralised matching system for kidney transplants. *Journal of the International Society for Telemedicine and eHealth*, 5:62–1. Cité page 77.