



**HAL**  
open science

# Service Chain Modeling and Embedding for NFV-based Content Delivery

Nicolas Herbaut, Daniel Negru, David Dietrich, Panagiotis Papadimitriou

► **To cite this version:**

Nicolas Herbaut, Daniel Negru, David Dietrich, Panagiotis Papadimitriou. Service Chain Modeling and Embedding for NFV-based Content Delivery. IEEE International Conference on Communications (ICC), May 2017, Paris, France. 10.1109/ICC.2017.7997165 . hal-01493381

**HAL Id: hal-01493381**

**<https://hal.science/hal-01493381>**

Submitted on 1 Jun 2021

**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.

# Service Chain Modeling and Embedding for NFV-based Content Delivery

Nicolas Herbaut\*<sup>‡</sup>, Daniel Negru\*, David Dietrich<sup>†</sup>, Panagiotis Papadimitriou<sup>†</sup>

\*Univ. Bordeaux, LaBRI, UMR 5800 F-33400 Talence, France {nherbaut,negru}@labri.fr

<sup>†</sup>Leibniz Universität Hannover, Germany {david.dietrich, panagiotis.papadimitriou}@ikt.uni-hannover.de

<sup>‡</sup>Viotech Communications SARL Versailles, France

**Abstract**—Increasing over-the-top video consumption endangers the sustainability of content delivery over the Internet. Internet Service Providers (ISP) face difficulties in competing on value-added services with content providers and Content Delivery Network (CDN) operators. In this respect, we propose a new model for the collaboration between content delivery stakeholders, so that CDN operators can deploy their software in ISP infrastructures leveraging on Network Function Virtualization (NFV). As the ISP network topology and utilization is deemed confidential, we use a high-level Service Level Agreement (SLA) for the negotiation of both computing resources and connectivity, allowing the ISP to optimize server selection, while providing at the same time sufficient flexibility to the CDN operators for content delivery. Furthermore, we present a linear programming formulation for the VNF Service Chain Embedding and an heuristic to increase problem tractability with a small cost overhead. Finally, we validate the efficiency of the proposed service chain model for virtual CDN management.

## I. INTRODUCTION

Media streaming, and video in particular, has become the most popular Internet service today. As the Internet was not originally designed for streaming high quality videos, delivering this massive volume of content is a challenging task. This has led to the increasing deployment of Content Delivery Networks (CDN) in order to improve application performance, while obviating the need for Content Providers (CP) to deploy their own infrastructure [1]. By deploying servers in strategic locations, CDN operators redirect client requests to proximate locations in order to achieve latency reduction and bandwidth conservation.

In this very competitive market, the majority of Internet Service Providers (ISP) is being kicked out of the video delivery value chain and is struggling to minimize the loss in their revenues [2]. However, even if their business role in media delivery is challenged by CDN operators, ISPs still hold a high valuable asset: they own and manage the network infrastructure.

To address this issue, we consider a collaboration opportunity between ISPs and CDN operators using Network Function Virtualization (NFV). More specifically, the proposed collaboration model enables ISPs to manage a distributed NFV infrastructure, whereas CDN operators have virtual CDNs (VCDN) deployed leveraging on NFV,

The work performed for this paper has been partially funded by the FP7 IP T-NOVA European Project (Grant Agreement N619520) and the FUI French National Project DVD2C

as exemplified in [3], [4]. This constitutes a *win-win* approach, as CDN operators can dynamically expand their coverage without the need to purchase and deploy new physical servers and ISPs can generate more revenue by leasing their infrastructure while reducing inter-domain traffic by serving content directly from their network.

To enable such a collaboration model, we face two main challenges. First, ISPs are known for their reluctance to disclose detailed information about their network topology to third parties (*e.g.*, CDN operators). To address this issue, we consider disclosing an overlay to CDN operators, granting them access and control through an API over virtual links and network functions for content distribution and delivery. In this respect, CDN operator requirements are expressed using a Service Level Agreement (SLA). The second challenge lies in embedding such SLAs into the ISP's network infrastructure [5]. We essentially express SLAs as Service Function Chains (SFC) and present a linear programming formulation for the SFC embedding problem. SFCs (contracted between the CDN operator and the ISP) are embedded offline by the ISP, prior to the service deployment. To facilitate the latter, we provide an API to the CDN operator for the management of the deployed services. As such, the CDN operator is granted with SLA guarantees as well as online management capabilities (to the extent of the deployed virtual network functions).

The remainder of the paper is organized as follows. Section II discusses the proposed collaboration model. Section III presents the SFC embedding problem formulation. In Section IV, we evaluate the gains achieved through such collaboration for offline SFC embedding and the online management capabilities offered to the CDN operator. Section V provides an overview of related work. Finally, Section VI concludes the paper and outlines future work.

## II. COLLABORATION MODEL

CDNs and ISPs do not naturally collaborate. They consider different optimization goals that are potentially in conflict with each other. [6]. In order to overcome these collaboration challenges, we propose a solution matching an ISP's connectivity "supply" with the CDN's connectivity "demand."

### A. High-Level Architecture

Our proposal aims at instantiating a NFV platform within the ISP Network capable of hosting, among others, virtual CDN (VCDN) services. In Figure 1, the ISP view shows two NFV Infrastructure Points of Presence (NFVI-

TABLE I: CDN-ISP SLA example

Name	Description	unit	example value	Simulation Hypothesis
Bit rate	Movie target bit rate	Mbps	2Mbps	randomly chosen between 2Mbps, 5Mbps and 10Mbps
Count	Movie download count	#	15000	randomly chosen from a normal distribution with mean 1000 and std dev 100
Content Duration	Movie average duration	minutes	90 min	randomly chosen from a normal distribution with mean 1h and std dev 10 min
SLA duration	Start and stop of the SLA	date range	From 00:00 01/01/2017 to 23:59 01/03/2017	randomly chosen from a normal distribution with mean 10h and std dev 1h
Client Group nodes (CG)	Identifiers for where the users connect	IP address /netmask	10.10.10.10/16	Random Nodes in the ISP Network
CDN Exit routes	CDN Peering point	IXP ID	netflix.franceix.net	Random Nodes in the ISP Network

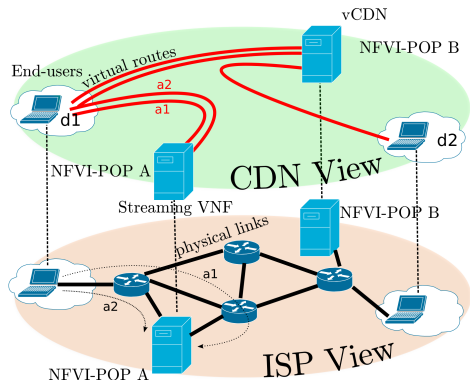


Fig. 1: VCDN deployed in ISP Network

POP) A and B deployed inside the ISP network. NFVI-POPs are data centres that provide computing resources (CPU, RAM, storage) for running Virtual Network Functions (vNFs), thus following a “Network Function as a Service” (NFaaS) approach. These NFVI-POPs are located close to the edge of the network where end-users (d1 and d2) are connected. In particular, NFVI-POPs can be connected with physical links and switches that provide support for Software Defined Networking (SDN), facilitating network management.

On the other hand, the CDN view abstracts all the underlying ISP topology. The CDN operators have only access to an overlay network that connects the end-users and the VCDNs servers. For example, virtual links a1 and a2 connect d1 with NFVI-POP A, while using different paths. This way, the underlying network is not disclosed to the CDN operators, maintaining the confidentiality of the ISP topology and resources, following a privacy-preserving “Connectivity as a Service” (CaaS) approach [7].

The current trend in NFV and SDN industry suggests that ISPs consider a massive invest in NFV and SDN technologies. A dozen of them are collaborating in open-source projects supporting those technologies, such as OpenDaylight. The perceived benefits are to promote technologies allowing combination of the orchestration of network and computing resources. ISPs can use it to build platforms supporting their core activity, but also to create a new revenue stream by selling them as a service competing on the cloud provider market.

Following this trend, ISPs are capable of contracting Service Level Agreements (SLA) with CDNs, enabling them to run their VCDN Network Functions. This brings significant advantages over traditional cloud providers’

services, as the ISP clients control the network end-to-end. Table I provides an example of an SLA. The CDN operator expresses a connectivity demand to the ISP which will be responsible for delivering it, if the SLA is accepted. The CDN operator does not know where the VCDN will be deployed, but it knows that the bandwidth and delay guarantees will be respected.

Once the SLA is established and the service deployed, the CDN uses a dedicated API to manage it. This API provides access to information needed to optimize the management of the virtual CDN, such as the residual bandwidth for each virtual route and the list of routed content from the VCDN and from the CDN peering point. Using this information and the API, the CDN can optimize content delivery according to his own business decisions. For example, a CDN can configure the system, so that each user gets its content from a VCDN server providing high quality of service, as long as its dedicated bandwidth limit is not reached. In the case where not all video streaming sessions can be delivered (for instance, after a burst of traffic), the CDN can reassign the delivery path from the VCDN link to the CDN peering point on its own network, making content delivery to be carried out in best effort mode. In this respect, several experiments were performed to demonstrate the benefits of this policy, as explained in Section IV-B.

#### B. Implementing the Content Delivery Service

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. In this section, we discuss the translation of the SLA into a Content Delivery Service Function Chain (SFC) request.

**The Canonical Service Model.** Once the CDN operator expressed the SLA, the underlying VCDN service can be represented under its canonical form. Starting from Client Groups (CG) (representing the geographical area toward End-Users), best effort connectivity  $B_{CDN}$  toward legacy CDN network and enhanced connectivity  $B_{VCDN}$  toward VCDN servers are depicted in Figure 2a

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 CDN AS. On the other hand, the  $CG \longleftrightarrow VCDN$  flow is targeted at an 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)

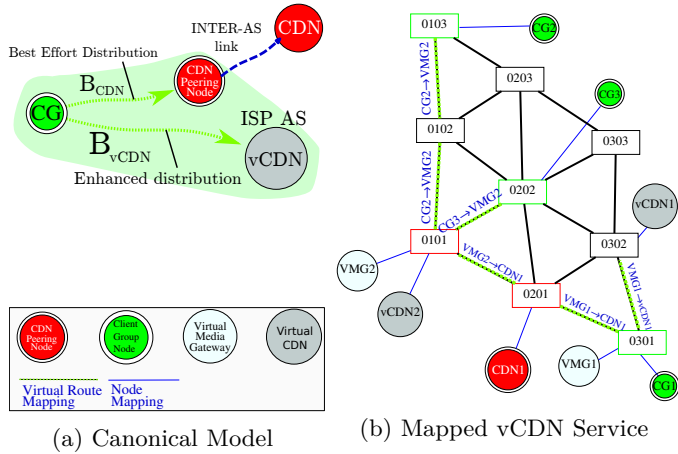


Fig. 2: Canonical and concrete mapping of a service chain. and is aimed to deliver higher quality videos.

**Leveraging Virtual Media Gateways.** Today, CDN operators use DNS redirection techniques [1] to specify which server will handle the client request. This application-layer method works with the granularity of domain names and is too coarse-grained to work at the *content* level. To mitigate this problem, we rely on Virtual Media Gateways (VMGs) to enhance content distribution. VMGs are transit virtual network functions that can inspect HTTP client requests and server responses to apply specific routing policies at flow level. ISPs have full control of their network and can leverage on SDN for more flexibility. SDN relies on APIs (*e.g.*, OpenFlow) that allow external controllers to populate flow entries on switches and routers. Currently, routing control using SDN is restricted to the lower layers (*i.e.*, OpenFlow does not support forwarding decisions based on the packet payload). This limitation can be overcome by coupling SDN with VMG. More specifically, we can steer traffic along a path (over SDN-enabled switching hardware) by inspecting the content of client requests at a VMG, while providing certain Quality of Service (QoS) guarantees.

**Service Chain Embedding.** With the addition of the VMG, the canonical model can now be reformulated as an SFC, as shown in Figure 2b. Subsequently, the ISP embeds this SFC generated from the CDN’s SLA into its network. This problem is known as SFC Embedding [5] and consists in mapping a service graph  $G^S = (N^S, E^S)$  to the ISP physical network graph  $G = (N, E)$ . Figure 2b shows such a mapping, where service nodes are mapped onto physical nodes and service edges are mapped onto physical edges. We formulate SFC embedding as an optimization problem, leveraging on existing work from the Virtual Network Embedding (VNE) literature [8], [9].

### III. SFC EMBEDDING PROBLEM

The implementation of the content delivery service, as discussed in Section II-B, consists of the embedding of SFCs by the ISPs. To this end, we introduce the cost function, the generation of service graphs and the linear model

TABLE II: Notations

Symbol	Domain	Description
<i>Domains and sets</i>		
$\mathcal{P}^{(i,j)}$	$N^S \times N^S \mapsto (E)^n$	set of the service edges linking $i$ to $j$ for $\mathcal{M}$
$\mathcal{D}^{i,j}$	$N^S \times N^S \mapsto \mathbb{N}$	delay between service nodes $i$ and $j$ for $\mathcal{M}$
$\mathcal{S}^{\mathcal{M}}$	$N^S$	set of CG nodes.
$\mathcal{V}_{CDN}$	$N^S$	set of vCDN nodes.
<i>Decision variables</i>		
$y_{u,v}^{i,j}$	$E \times E^S \mapsto \{0, 1\}$	$\begin{cases} 1, & \text{if } (i, j) \text{ is mapped } (u, v) \\ 0, & \text{otherwise} \end{cases}$
$x_u^i$	$N \times N^S \mapsto \{0, 1\}$	$\begin{cases} 1, & \text{if } u \text{ hosts service node } i \text{ for } \mathcal{M} \\ 0, & \text{otherwise} \end{cases}$
<i>Physical Topology and Service parameters</i>		
$d_{u,v}$	$E \times E \mapsto \mathbb{R}^+$	the delay for edge $(u, v)$ of node
$d_i$	$N^S \mapsto \mathbb{R}^+$	the delay for edge $(u, v)$ of node
$b_{i,j}^S$	$E^S \times E^S \mapsto \mathbb{R}^+$	required bandwidth between $i$ and $j$ .
$b_{u,v}$	$E \times E \mapsto \mathbb{R}^+$	available bandwidth between $u$ and $v$
$c_i^S$	$N^S \mapsto \mathbb{R}^+$	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

assuring that the SLA constraints are fulfilled within the embedding process.

#### A. Optimization Problem Formulation

A valid mapping, as depicted in Figure 2b, fulfills all the SLA requirements. To this end, we formulate the optimization problem using common network-flow notations, summarized in Table II. We use two main binary decision variables,  $x_u^i$  to denote that the service node  $i$  is deployed on NFVI-POP  $u$  and  $y_{u,v}^{i,j}$  to denote that physical link  $(u, v)$  is used to support all or part of the service edge  $(i, j)$ .

First, we must ensure (1) that each service node is mapped to an NFVI-POP

$$\sum_{u \in N} x_u^i = 1, \forall i \in N^S \quad (1)$$

Contrary to traditional hardware devices, VNFs are elastic and can adapt to varying workloads by automatically extending their footprint with to scale-out and scale-in features. Augmenting the capacity of a VNF requires increasing the number of its basic building blocks, also known as Virtual Network Function Components (VNFC). 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 VM. In this paper, we assume that each VNF is dimensioned from the start to the maximum capacity needed to fulfil the SLA, thus requiring  $c_i^S$  VM from the NFVI-POP where it is mapped. (2) makes sure that the NFVI-POP  $u$  can host the number of VMs required by each VNF  $i$  it hosts.

$$\sum_{i \in N^S} x_u^i \times c_i^S \leq c_u, \forall u \in N \quad (2)$$

Each service edge is possibly supported by several physical links, so by applying (3), we make sure that topology bandwidth  $b_{u,v}$  can satisfy the demand for each mapped service edge  $b_{i,j}$ .

$$\sum_{(i,j) \in E^S} y_{u,v}^{i,j} \times b_{i,j}^S \leq b_{u,v}, \forall (u, v) \in E \quad (3)$$

Delay constraints (4) assure that  $\forall s \in \mathcal{S}$  the CGs, and  $\forall t \in \mathcal{T}$  the VCDNs, we do respect the maximal end-to-end delay  $\mathcal{D}_{\mathcal{M}}(s, t)$  on each path  $\mathcal{P}_{\mathcal{M}}^{(s,t)}$  joining CG to

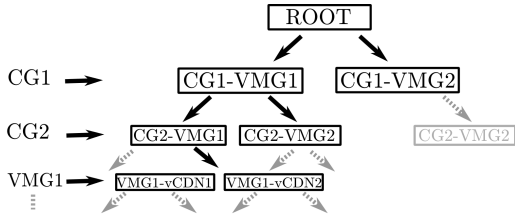


Fig. 3: Building a tree with Service Edges

VCDN, taking both physical links transmission delay  $d_{u,v}$  and service node processing delay  $d_i$  into account.

$$\sum_{(i,j) \in \mathcal{P}_{\mathcal{M}}^{(s,t)}} \sum_{(u,v) \in E} y_{u,v}^{i,j} \cdot d_{u,v} + x_u^i \cdot d_i \leq \mathcal{D}_{\mathcal{M}}(s,t) \quad (4)$$

We also make sure that the service does not loop on the physical topology (5) by assuring that no more than one outbound link  $(u,v)$  is present for the same service edge  $(i,j)$ .

$$\sum_{v \in \delta^+(u)} y_{u,v}^{i,j} \leq 1, \forall (i,j) \in E^S, \forall u \in N \quad (5)$$

Finally, we apply the flow conservation constraint (6).

$$\sum_{v \in N} y_{u,v}^{i,j} - y_{v,u}^{i,j} = x_u^i - x_u^j, \forall (i,j) \in E^S, \forall u \in N \quad (6)$$

We try to tackle nodes and edges assignments at the same time, so the problem is known to be  $\mathcal{NP}$ -hard [8]. In the next subsection, we detail the cost function and the generation of the service graphs according to it.

#### B. The cost function

We aim at minimizing the total cost of embedding for a mapping  $\mathcal{M}_S$  subject to constraints (1)-(6) derived from the SLA  $S$ . Each ISP has a different cost structure, however, in this paper, we consider the 2 main costs: networking  $p_{net}(\mathcal{M}_S)$  and hosting costs the VNF  $p_{vnf}(\mathcal{M}_S)$ . We can write the cost function as (7).

$$p(\mathcal{M}_S) = p_{net}(\mathcal{M}_S) + p_{vnf}(\mathcal{M}_S) \quad (7)$$

We assume that the network cost is proportional to the consumed bandwidth on all enhanced service paths (CG-VCDN) so (7),

$$p_{net}(\mathcal{M}_S) = \sum_{(u,v) \in E} \left( \sum_{(i,j) \in E^S, j \notin CDN} y_{u,v}^{i,j} * b_{i,j} \right) \quad (8)$$

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  $c_i(\cdot)$  of the bandwidth transiting through or targeted at them.

$$p_{vnf}(\mathcal{M}_S) = \sum_{j \in N^S} c_j \left( \sum_{i \in \delta^+(j)} b_{i,j} \right) \quad (9)$$

In the next section, we describe the computation of bandwidth in the service graph generation process.

#### C. Generating the Service Graphs

In the previous section, we let the cost function use edge bandwidth  $b_{i,j}$  and VNF cost  $c_i$  as decision variables, which makes 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. In this paper, we consider that each Client Group

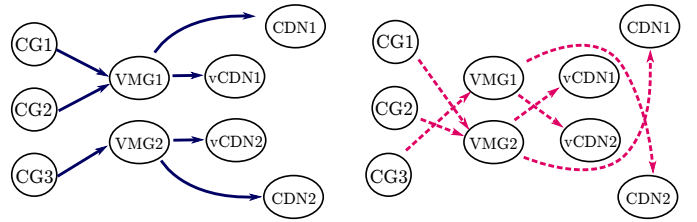


Fig. 4: 2 Service-isomorphic Graphs

is connected to exactly 1 VMG, and that each VMG is connected exactly to 1 VCDN and 1 CDN. To build each possible Service Graph, we first create the service nodes (VMG, VCDN, CDN peering point) and enumerate all the possible service edges combinations by constructing a tree which nodes are all the possible edges of the service graph (Figure 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. This process generates a very high number of possible services, let us now consider a method to reduce it only to meaningful ones. 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 VCDN) 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 4 are equivalent when renaming un-mapped nodes, i.e., VMG1 to VMG2 and VCDN1 to VCDN2 and vice-versa. We give a more formal definition 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 VCDN).

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 consider during the optimization phase:

**Definition 2** For a given SLA  $S$ , a *Full Service Graph Class*  $\mathcal{C}_S(n, m)$  is the set of all the connected service graphs having up to  $n$  VMGs and up to  $m$  VCDNs, 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 (7), which becomes linear with respect to service graphs. Solving the embedding problem is finally done by computing the optimal mapping for each service graphs, and selecting the cheapest among them.

#### D. Heuristic approach to compute Service Graphs

As above mentioned, the ILP is by itself NP-Hard. Due to the necessary optimisation of a large number of service graphs in  $\mathcal{C}_S(n, m)$ , optimization becomes quickly

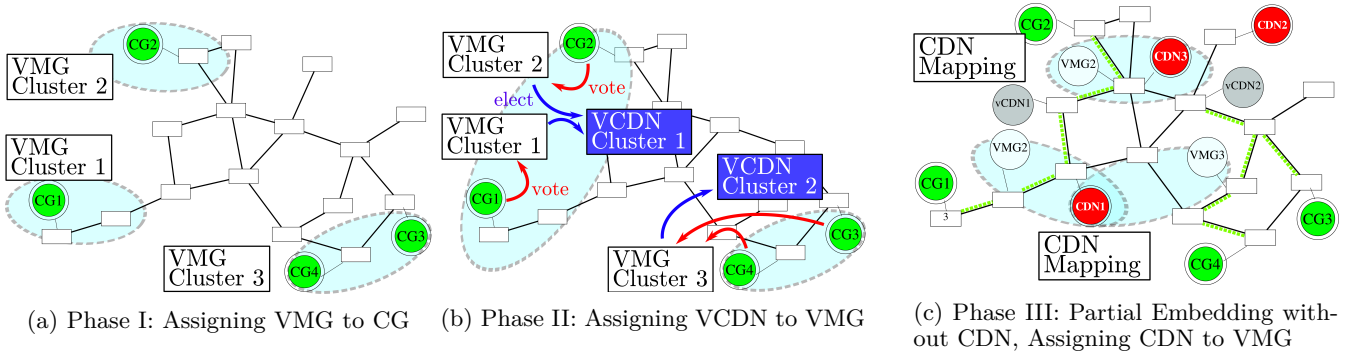


Fig. 5: The tree steps of Heuristic Service Graph Generation

TABLE III: # of Services graphs and computation time

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

TABLE IV: Cost Hypotheses

VMG	1 VM + 1 VM / Gbps @ 20\$/VM/h
VCDN	20 VM + 1 VM / Gbps @ 40\$/VM/h
Network	0.63\$/Mbps/month

intractable for 5 CGs or more. To circumvent this issue, we generate a much smaller Reduced Service Graph Class  $\hat{\mathcal{C}}_S(n, m)$  with a service topology generation heuristic that uses the characteristics of the physical topology to decide how to generate service edges efficiently. We now present a three-steps 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 5a. We repeat the procedure in Phase II, Figure 5b, by creating VCDN Clusters that, for a given number of VCDNs, minimize the intra-cluster delay between CGs. Note that the number of VCDN 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 VCDN nodes. In the last Phase III, we compute  $\tilde{G}^S$  delays and bandwidths, neglecting the CDN edges. We use the ILP to generate a partial mapping  $\mathcal{M}$ , which provides estimates for VMG and VCDN node mapping. With those estimates, we are able to determine for each VMG node, which CDN peering point is the closest, as shown in Figure 5c. Based on this information, we finally create the service edges between VMGs and CDNs and compute  $\hat{G}^S$ . We save  $\mathcal{M}$ , as it will be used by the re-optimization feature of our ILP tool-kit to compute the full mapping  $\mathcal{M}$  faster. In the next section, we compare the performance and computation time of this heuristic vs the Full Service Graph.

## IV. EVALUATION

### A. Offline SFC embedding

To evaluate the proposed solution, we developed a simulation environment that generates services according to the SLA, optimizes them with our ILP solver and finally selects the best embedding<sup>1</sup>. We carried out our simulations on a real topology retrieved from an open database [10]. 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 VCDN is dimensioned according to the served bandwidth, based on lab measurements on our prototype implementation conducted within the T-NOVA project [?]. 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 needed to generate the services of in the Full and Reduce Service Graph Classes, then we compare the prices obtained by embedding the services on the physical topology.

#### 1) Service Graph Generation

Table III shows the computation time results that we obtained on a Laptop equipped with an Intel i7-4600U CPU @ 2.10GHz, after generating the services graphs for an SLA composed of 4 CGs and 2 CDNs peering points. We compare the generation time for both the full and the reduce service graph classes. For the Reduce 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 re-

<sup>1</sup>Sources and online demo: [www.labri.fr/~nherbaut/icc2017.html](http://www.labri.fr/~nherbaut/icc2017.html)

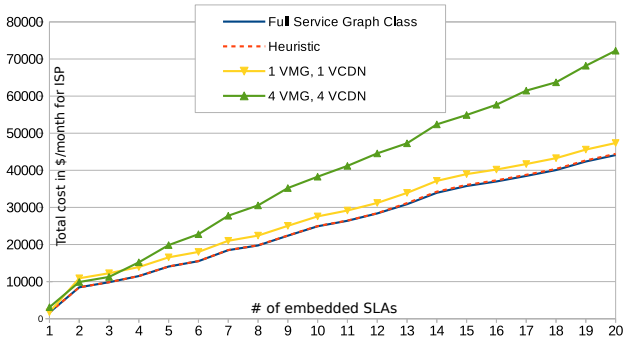


Fig. 6: Cost Comparison of serie of successive embedding performed on the Geant Topology

ducing 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 as the heuristic uses the SLA and the physical topology, it is sensible to the size of physical topology graph. So computing the service graph on the Reduced class for a larger graph increases the computation time, which is not the case for 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  $R_{58}^{20}$ . In future work, we plan to consider other methods than the ILP for the embedding to further improve the scalability of our model.

## 2) Embedding quality comparison

For the experimentation, we used the SCIP tool-kit to generate mappings. 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 [11].

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 IV summarize the cost hypotheses. Our simulation results are shown in Figure 6. For comparison, we also included trivial services : (A) with 1 VMG and 1 VCDN and (B) with 4 VMGs and 4 VCDNs. The results obtained by optimizing on the Reduced Service Graph class are very close to the result 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 by the heuristic. The heuristic has also lower cost than trivial services (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.

## B. Online VCDN management

To illustrate the management phase of the SFC instance by the CDN, we have implemented a virtual test-bed with Mininet, shown in Figure 7. The service contains a CG Node, a VMG, a VCDN and a CDN. We have generated 200 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 divided into two classes, *i.e.*, the users (25% of the total) accessing paid content and the users accessing free content.

The CDN has created two different configurations: in the first one, every HD session is routed to the VCDN, since it provides guaranteed QoS. SD sessions are still routed to the CDN's own network outside the ISP AS, delivering content in best effort. In the second configuration, only paying contents are served by the VCDN. This second configuration is activated automatically whenever the bandwidth used on the VMG-VCDN link is greater than 80% of the available bandwidth. The first configuration is reactivated when the used bandwidth of the VCDN link goes below 50% of the bandwidth requested in the SLA. To avoid oscillations between configuration 1 and 2, we used a rolling average of 25s window. Updating the configuration and reading bandwidth statistics is done through a REST API exposed by an SDN controller implemented on top of the Ryu Framework.

The results of this experiment are shown in Figure 7. It can be seen that the VCDN link starts to reach its limit at 50s. The CDN's network configuration mechanism detects this saturation at 75s (due to the rolling average), and installs configuration 2. At this point, the free HD traffic is routed to the CDN, causing a drop in the VCDN traffic and an increase in the CDN link. Thanks to the responsiveness of the network control mechanism, only a few buffer starvations are reported at the very beginning of the traffic burst (buffer starvation causes playback freeze in video which degrades QoE). Once the quantity of traffic on the VCDN decreases, configuration 1 is reactivated, enabling every HD user to benefit from the quality of the VCDN link again.

## V. RELATED WORK

In this section, we discuss related work on service chain embedding. Most existing work maps NF-graphs onto a single substrate network, which is usually assumed to be a data centre network (*e.g.*, with a fat-tree topology). In this respect, authors in [12] investigate gains in terms 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. [13] 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. [14] present a solution to the NF-graph mapping problem,

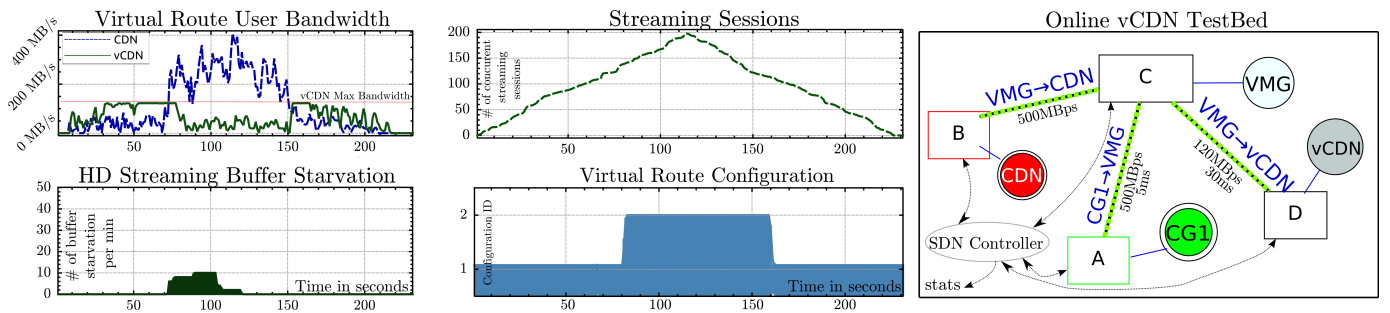


Fig. 7: Online VCDN management results.

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 [15] 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 [16], 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. [17] derive an ILP formulation and heuristic algorithm for service mapping with the objective of operational cost and resource fragmentation minimization.

Compared to these SFC embedding approaches, we propose a SFC embedding method that accepts elastic services, for which the exact service characteristics are computed from an SLA. In addition, we provide built-in support for CDN specific requirements such as SLA-driven delay, bandwidth constraints, AS interconnections.

## VI. CONCLUSIONS AND FUTURE WORK

We proposed a framework for ISP-CDN collaboration, permitting CDNs to run their content delivery functions in a much more optimized way at all levels (investments, deployment, maintenance, costs). The problem of embedding a Content Delivery SFC over the ISP network was modelled, leveraging VNFs, such as Virtual Media Gateways. We solved the problem with linear programming, by evaluating on all the possible services graph to find the best mapping. By providing a heuristic that use the physical topology characteristics, we can generate only a subset of the possible services graph which not only make the problem tractable, but also shows little impact on final embedding cost. We concluded by experimenting our implementation on an SDN testbed, to demonstrate the manageability of the vCDN service. In future work, we plan do address dynamic SLAs, survivable embedding, improve the service generation algorithm and embedding scalability.

## REFERENCES

- [1] M. Pathan, "Cloud-based content delivery and streaming," *Advanced Content Delivery, Streaming, and Cloud Services*, 2014.
- [2] J. Wulf, R. Zarnekow, T. Hau, and W. Brenner, "Carrier activities in the CDN market - an exploratory analysis and strategic implications," in *2010 14th International Conference on Intelligence in Next Generation Networks (ICIN)*, 2010.
- [3] N. Bouten, J. Famaey, R. Mijumbi, B. Naudts, J. Serrat, S. Latre, and F. De Turck, "Towards NFV-based multimedia delivery," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 738–741.
- [4] N. Herbaut, D. Négru, Y. Chen, P. A. Frangoudis, and A. Ksentini, "Content delivery networks as a virtual network function: a win-win isp-cdn collaboration," in *IEEE Global Communications Conference*, 12/2016 2016.
- [5] P. Quinn and T. Nadeau, "Problem statement for service function chaining, rfc 7498," 2015.
- [6] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang, "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*, ser. SIGMETRICS '09. ACM, 2009, pp. 239–250.
- [7] Y. Wang, Q. Hu, and X. Cao, "Connectivity as a service: Towards optical-based network virtualization," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*. IEEE, 2014, pp. 264–268.
- [8] M. Chowdhury, M. R. Rahman, and R. Boutaba, "Vineyard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 1, pp. 206–219, 2012.
- [9] A. Fischer, J. Botero, M. Till Beck, H. de Meer, and X. Hesselbach, "Virtual Network Embedding: A Survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, 2013.
- [10] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, 2011.
- [11] T. Achterberg, "Scip: solving constraint integer programs," *Mathematical Programming Computation*, vol. 1, no. 1, 2009.
- [12] S. Sahhaf, W. Tavernier, M. Rost, S. Schmid, D. Colle, M. Pickavet, and P. Demeester, "Network service chaining with optimized network function embedding supporting service decompositions," *Comput. Netw.*, vol. 93, pp. 492–505, Dec. 2015.
- [13] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 1346–1354.
- [14] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *CloudNet, 2014 IEEE 3rd International Conference on*, Oct 2014, pp. 7–13.
- [15] T. Lukovszki, M. Rost, and S. Schmid, "It's a match!: Near-optimal and incremental middlebox deployment," *SIGCOMM Comput. Commun. Rev.*, vol. 46, no. 1, pp. 30–36, Jan. 2016.
- [16] T. Lukovszki and S. Schmid, "Online admission control and embedding of service chains," in *Structural Information and Communication Complexity*. Springer, 2014, pp. 104–118.
- [17] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On orchestrating virtual network functions in NFV," *CoRR*, 2015.