Optimal Planning of Virtual Content Delivery Networks under Uncertain Traffic Demands

Michele Mangili[a], Jocelyne Elias[b], Fabio Martignon[c], Antonio Capone[d]

[a]*LRI, Université Paris-Sud & DEIB, Politecnico di Milano*
[b]*LIPADE, Université Paris Descartes & INRIA*
[c]*LRI, Université Paris-Sud & IUF, Institut Universitaire de France*
[d]*DEIB, Politecnico di Milano*

Abstract

Content Delivery Networks (CDNs) have been identified as one of the relevant use cases where the emerging paradigm of Network Functions Virtualization (NFV) will likely be beneficial. In fact, virtualization fosters flexibility, since on-demand resource allocation of virtual CDN nodes can accommodate sudden traffic demand changes. However, there are cases where physical appliances should still be preferred, therefore we envision a mixed architecture in between these two solutions, capable to exploit the advantages of both of them. Motivated by these reasons, in this paper we formulate a two-stage stochastic planning model that can be used by CDN operators to compute the optimal long-term network planning decision, deploying physical CDN appliances in the network and/or leasing resources for virtual CDN nodes in data centers. Key findings demonstrate that for a large range of pricing options and traffic profiles, NFV can significantly save network costs spent by the operator to provide the content distribution service.

Keywords: *Content Distribution, Network Function Virtualization, Resource Allocation, Stochastic Network Planning.*

Email addresses: michele.mangili@lri.fr (Michele Mangili), jocelyne.elias@parisdescartes.fr (Jocelyne Elias), fabio.martignon@lri.fr (Fabio Martignon), antonio.capone@polimi.it (Antonio Capone)

1. Introduction

The worldwide success of content-rich web applications like social networks or on-demand streaming services has forced network operators to invest a significant amount of money in order to keep up-to-date their communication infrastructures [1]. In particular, Content Delivery Networks (CDNs) have nowadays become a necessary (and well-established) technology to efficiently serve the traffic demands that consumers are generating, while supporting the high level of performance and reliability that providers are demanding [2].

Although CDN is an effective infrastructure to move replicas of popular contents closer to the users' locations, it requires significant investments to be built and operated. As an example, the Akamai infrastructure comprises more than 61 000 servers deployed in 1 000 networks and 70 countries worldwide [3]. To reduce the capital expenditures and improve the performance of CDNs, organizations such as the Internet Engineering Task Force (IETF) and the European Telecommunications Standards Institute (ETSI) have recently begun a standardization process for two alternative architectures:

- Content Delivery Network Interconnection (CDNI);
- Virtual Content Delivery Network (vCDN).

Despite the fact that they both have to deal with network content distribution, these proposals have a radically different scope: the former (CDNI) is mostly concerned with the co-operation of many CDN providers [4], whereas the latter (vCDN) proposes to virtualize the CDN services on top of the novel layer for Network Functions Virtualization (NFV) [5].

While both the architectures aim at optimally exploiting available physical resources, the grounds of CDNI are settled on agreements between different CDN operators that often are in direct competition in the same market. On the other hand, the NFV approach is to run network functions in a virtualized environment, executed on a shared physical infrastructure composed of industry standard high volume servers, storage and switches [5]. Therefore, vCDN implemented on top of NFV enjoys the positive advantage of avoiding potential competition issues, since the virtualized environment ensures the necessary level of isolation between the different network functions. Furthermore, spare NFV substrate capacity can be leased by network operators to third parties, a condition that makes vCDN appear even more profitable.

Motivated by the previous background, in this paper we tackle a fundamental issue that arises in such context: the planning problem for a mixed physical-virtual Content Delivery Network under uncertain traffic demands. In our formulation, the CDN operator can choose between *purchasing* physical CDN appliances and *leasing* instances of virtual CDN nodes provided by an infrastructure operator. However, while vCDN nodes can be activated on-demand if the traffic requests require to do so, the installation of physical CDN nodes must be chosen on a long-term schedule. For both physical or virtual CDN surrogate servers, the operator must carefully choose their location, while minimizing the overall costs. Due to the fact that planning is performed on a long-term basis, the theoretical framework of *stochastic optimization* will be used to guarantee robustness of the solution with respect to the uncertainty in the probabilistic description of future traffic demands.

The contributions of this paper are summarized as follows:

1. We formulate a two-stage stochastic planning model used by CDN operators to compute optimal, long-term network planning decisions, under traffic demands uncertainty. In particular we introduce the deterministic equivalent program in the extensive form of the two-stage stochastic CDN planning program.

2. We propose a greedy heuristic approach that finds good planning solutions (close to the optimum, in several cases) in polynomial time even for large-scale network topologies.

3. We implement the *single* and *multicut* versions of the L-shaped algorithm [6], which is a very effective approximation method for the two-stage stochastic problem we formulate in this work.

4. We perform an extensive numerical evaluation, considering real scale topologies and a wide range of parameters. We further compare the execution time of the greedy heuristic with the proposed *exact* solution strategies: (1) the deterministic equivalent program and (2) the L-shaped algorithm.

Our key findings suggest that a mixed physical-virtual CDN infrastructure leads to significant lower costs when compared to those obtained by a standard CDN, while being robust with respect to sudden traffic demand changes.

The paper is organized as follows: Sec. 2 discusses related work. In Sec. 3 we present our contribution; in particular, the system is described in Sec. 3.1; the optimization model is presented in Sec. 3.2, while the L-shaped and greedy algorithms are illustrated in Sec. 3.3. Numerical results are presented in Sec. 4, and Sec. 5 concludes this paper.

2. Related Work

In this section, we survey relevant literature on Network Functions Virtualization (Sec. 2.1), Content Delivery Networks (Sec. 2.2) and Stochastic Optimization techniques (Sec. 2.3).

2.1. Network Functions Virtualization

One of the key enabling paradigms that will considerably increase the dynamicity of ICT networks is Network Functions Virtualization (NFV) [5], which is discussed in recent surveys [7, 8, 9, 10, 11]. Indeed, Service Providers and Network Operators are facing increasing problems to design and implement novel network functionalities, following the rapid changes that characterize the current Internet and Telecom operators [12].

To support the network virtualization paradigm, one of the challenges that must be solved is to find a mapping between a set of requests for virtual network resources and the available underlying physical infrastructure, ensuring that desired performance requirements on nodes and links are guaranteed [13]. This is the *virtual network embedding* (VNE) problem, which is known to be NP-hard, since it can be reduced to the multi-way separator problem [14]. VNE has received a lot of attention from the community, and several heuristic algorithms have been proposed, e.g. in [15, 16, 17, 18].

Jarray et al. propose in [15] a column-generation technique coupled with a rounding heuristic to discover the most profitable embedding, under the constrained physical capacity of the infrastructure. Deterministic and randomized rounding techniques are used by Chowdhury et al. in [16], where they further facilitate the virtual link mapping by designing an augmented graph description to efficiently support node location constraints. Cheng et al. in [17] solve the node mapping step with a greedy algorithm: higher ranking is given to the nodes that possess more spare resources and are placed in better locations of the network.

Rather than assuming that the operator knows *a-priori* the traffic demands, our contribution is to consider the case where their probability distribution is known, and our proposed formulation is robust with respect to such uncertainty.

2.2. Content Delivery Networks

In the last few years, *content multihoming* is emerging as a novel technique for content delivery networks that makes it possible to jointly use many CDN services: [1, 2, 19, 20].

Adhikari et al. show in [1] that the Netflix infrastructure already leverages multiple CDNs (Akamai, LimeLight and Level-3). The authors observe that the customers are mapped to a particular CDN in a rather static manner. In [2], Liu et al. further confirm that other major content publishers such as Hulu, Microsoft, Apple, Facebook and MSNBC are currently already exploiting content multihoming. Furthermore, given the practical relevance of this architecture, they design optimization algorithms to minimize the overall distribution costs under constrained quality requirements. Finally, Wang et al. extended in [19] the work of Liu et al., by explicitly considering capacity constraints on the surrogate nodes.

The novelty of our approach is to consider a new distribution architecture, implemented on top of NFV, where virtual CDN nodes can be used for content delivery purposes. Rather than forcing the provider to settle agreements with other CDN operators, our proposal guarantees better isolation and dramatically limits potential competition issues.

2.3. Stochastic Optimization

In practical scenarios, network design cannot assume that future traffic demands are known a-priori; on the other hand, more advanced optimization techniques must be used to take into consideration the stochastic nature of input parameters: [21, 22, 23].

Atamtürk et al. formulate in [21] a two-stage network design model with traffic demand uncertainty. In their approach, the operator performs the planning decision according to a probabilistic description of traffic demands. The value of the second-stage recourse variables is chosen by changing flow routing. Liu describes in [22] the basic stochastic procedures applied to a flow assignment network design problem, showing the *here-and-know* solution and the *scenario-tracking* result obtained for the flow-assignment. A multistage stochastic programming model for mobile radio access networks has been proposed by Eisenblätter et al. in [23]. In their formulation they jointly take into consideration the coverage and capacity of their communication infrastructure.

In line with previous literature, our formulation takes into account the uncertainty embedded in future traffic demands. Our contribution is to apply the theoretical framework of stochastic optimization to content distribution.

3. Optimal Content Delivery in NFV

In this section we describe our proposed solution for the optimal content delivery planning in NFV. Sec. 3.1 introduces the system model and relevant assumptions. In Sec. 3.2 we formulate the optimization model, while in Sec. 3.3 we discuss the design of the L-shaped and greedy algorithms.

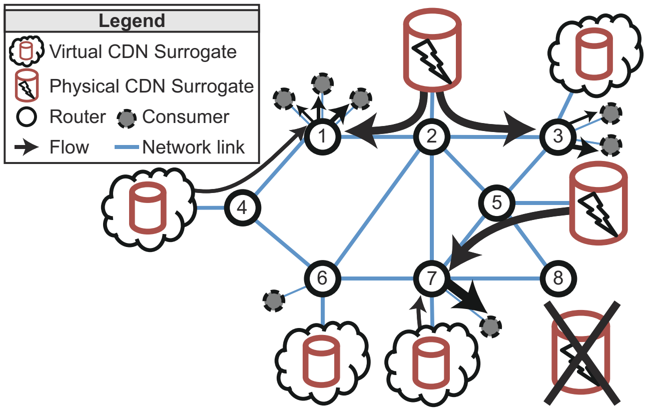


Figure 1: *System model.* The network is composed by consumers, routers, virtual and physical CDN surrogates. Our proposed optimization model selects (1) the planning of physical CDNs and (2) request routing.

3.1. System Model and Assumptions

Figure 1 shows the system model we consider in our proposal. In this work we tackle the long-term planning problem from the point of view of a CDN provider. The aim of the provider is to perform two choices:

1. Select *whether* and *where* physical CDN nodes should be installed in the network topology;
2. Select the optimal *request routing*, given the installed physical CDNs and the virtual nodes available.

Since the planning decision is operated on a long-term time schedule, the provider does not deterministically know what is going to happen in the future. On the other hand, we assume that an estimate of the continuous probability distribution of future traffic demands is known for the planning problem. However, for the sake of simplicity, and as frequently done in the literature (e,g: [6, 23]), we discretize this information on a finite number of scenarios, therefore our traffic model is jointly *time-varying* and *stochastic* in its nature.

As shown in Fig. 1, traffic demands are expressed by the consumers. Virtual and physical CDN nodes can both be used to efficiently serve consumers' demands, however there exist major differences (in terms of capacity, activation choice and pricing policy) between these two types of CDN hosts:

- *Capacity:* virtual CDN nodes can serve a lower amount of traffic requests since the presence of the hypervisor and the shared hardware infrastructure reduces the throughput of the CDN surrogates.

- *Activation choice:* virtual CDN servers can be used on-demand, and they do not need to be explicitly activated. On the other hand, if the operator chooses to install a physical CDN server, it will be activated once and it will stay active throughout the entire time horizon.

- *Pricing policy:* physical CDN nodes have an activation price that considers both the capital expenditure (CAPEX) for the acquisition of the device as well as the long-term operational expenditure (OPEX) costs. On the other hand, the virtual CDN nodes have a traffic-proportional price related to the OPEX cost component, which is the per-bandwidth leasing price that the vCDN owner charges.

Since virtual CDN nodes can be used on-demand, they can serve the portion of traffic requests with the highest variability. To improve the quality of service of a CDN, surrogate servers must be selected close to the location of consumers. For this purpose, we use the link delay to control the performance of the infrastructure: we assume that the content provider wants to serve a fraction of the overall requests within a bounded limit on the delay.

Popular CDN providers such as Amazon CloudFront or Microsoft Azure CDN do not have an activation cost but charge for their services according to the amount of traffic that surrogates are providing, regardless of the caching storage used. Moreover, frequent flash crowds make the object popularity suddenly change, whereas having an estimate of the aggregate future demands is instead much easier [24]. For these reasons, we focus on infrastructure planning and request routing, while we do not tackle the replica placement problem.

Table 1: Notation used in this paper.

Input Parameters	
$\mathcal{D}$	Set of consumers (*destination* nodes)
$\mathcal{S}$	Set of candidate surrogate servers (*source* nodes) $\mathcal{S} = \mathcal{S}^P \cup \mathcal{S}^V$
$\mathcal{S}^P$	Set of candidate physical CDN servers
$\mathcal{S}^V$	Set of candidate virtual CDN servers
$\mathcal{T}$	Set of time slots
Φ	Set of stochastic scenarios
$r_d^{t,\phi}$	Traffic requests of client $d \in \mathcal{D}$, at time slot $t \in \mathcal{T}$, for scenario $\phi \in \Phi$
ϵ	Minimum service level guaranteed (fraction of traffic requests served with a bounded delay of at most Δ)
Δ	Maximum tolerated delay
$\delta_{s,d}$	Delay between the nodes $s \in \mathcal{S}$, $d \in \mathcal{D}$
$\mathcal{K}_s^P$	Capacity of the physical CDN server $s \in \mathcal{S}^P$
$\mathcal{K}_s^V$	Capacity of the virtual CDN server $s \in \mathcal{S}^V$
$\mathcal{C}_s^P$	CAPEX and OPEX costs of the physical CDN server installed at a candidate site $s \in \mathcal{S}^P$
$\mathcal{C}_s^V$	Usage cost of the virtual CDN server at the candidate site $s \in \mathcal{S}^V$
p_ϕ	Realization probability for the scenario $\phi \in \Phi$

Decision Variables	
a_s	0-1 Physical CDN activation variable. $a_s = 1$ if a physical CDN is installed at the candidate point $s \in \mathcal{S}^P$
$y_{s,d}^{t,\phi}$	Physical CDN flow variable for requests served by $s \in \mathcal{S}^P$ to client $d \in \mathcal{D}$, at time $t \in \mathcal{T}$, and scenario $\phi \in \Phi$
$z_{s,d}^{t,\phi}$	Virtual CDN flow variable for requests served by $s \in \mathcal{S}^V$ to client $d \in \mathcal{D}$, at time $t \in \mathcal{T}$, and scenario $\phi \in \Phi$

3.2. Optimization Model

In this section we describe the optimization model we formulate for the optimal planning of a mixed physical-virtual CDN infrastructure. The notation is summarized in Table 1.

Let $\mathcal{S} = \mathcal{S}^P \cup \mathcal{S}^V$ be the set of CDN surrogate servers, where $\mathcal{S}^P$ and $\mathcal{S}^V$ represent candidate physical and virtual surrogate nodes, respectively. Consumers are denoted with $\mathcal{D}$, the set of time slots is represented with $\mathcal{T}$, while the set of stochastic scenarios is Φ. Each scenario $\phi \in \Phi$ has an associated realization probability, represented by p_ϕ. Consumers $d \in \mathcal{D}$ express a time-varying traffic demand for each scenario $\phi \in \Phi$, that we indicate with $r_d^{t,\phi}$. The CDN provider ensures that at least a fraction ϵ of the aggregate requests in every time slot is served by CDN nodes within a bounded delay, denoted as Δ. The topological information is encoded in our proposed optimization model using the $\delta_{s,d}$ input parameter, which represents the delay between client $d \in \mathcal{D}$ and CDN node $s \in \mathcal{S}$. $\mathcal{K}_s^P$ and $\mathcal{K}_s^V$ are the bandwidth capacities for physical and virtual CDNs, respectively. Physical nodes have an activation cost $\mathcal{C}_s^P$, while virtual CDN nodes have a traffic-proportional cost $\mathcal{C}_s^V$.

Our proposed optimization model chooses the optimal physical nodes placement and request routing. a_s is a binary decision variable that is set to 1 if and only if the physical candidate server $s \in \mathcal{S}^P$ is activated. Traffic requests for consumer d, in time slot t for scenario ϕ can be served by flows $y_{s,d}^{t,\phi}$ and $z_{s,d}^{t,\phi}$. In particular, $y_{s,d}^{t,\phi}$ is a flow originating from the physical node $s \in \mathcal{S}^P$, while $z_{s,d}^{t,\phi}$ is a flow provided by the virtual node $s \in \mathcal{S}^V$.

The deterministic equivalent program in the extensive form for the CDN planning problem of our infrastructure (EF-CDN) is formulated as follows:

$$\min \sum_{s \in \mathcal{S}^P} \left[\mathcal{C}_s^P a_s \right] + \mathbb{E}_\Phi \left[\sum_{s \in \mathcal{S}^V} \sum_{t \in \mathcal{T}} \sum_{d \in \mathcal{D}} \left(\mathcal{C}_s^V z_{s,d}^{t,\phi} \right) \right] \tag{1}$$

subject to:

$$\sum_{d \in \mathcal{D}} y_{s,d}^{t,\phi} \leq a_s \mathcal{K}_s^P \qquad \forall s \in \mathcal{S}^P, t \in \mathcal{T}, \phi \in \Phi \tag{2}$$

$$\sum_{d \in \mathcal{D}} z_{s,d}^{t,\phi} \leq \mathcal{K}_s^V \qquad \forall s \in \mathcal{S}^V, t \in \mathcal{T}, \phi \in \Phi \tag{3}$$

$$\sum_{s \in \mathcal{S}^P} y_{s,d}^{t,\phi} + \sum_{s \in \mathcal{S}^V} z_{s,d}^{t,\phi} = r_d^{t,\phi} \qquad \forall d \in \mathcal{D}, t \in \mathcal{T}, \phi \in \Phi \tag{4}$$

$$\frac{\sum\limits_{d \in \mathcal{D}} \left[\sum\limits_{s \in \mathcal{S}^P | \delta_{s,d} \leq \Delta} y_{s,d}^{t,\phi} + \sum\limits_{s \in \mathcal{S}^V | \delta_{s,d} \leq \Delta} z_{s,d}^{t,\phi} \right]}{\sum\limits_{d \in \mathcal{D}} r_d^{t,\phi}} \geq \epsilon \qquad \forall t \in \mathcal{T}, \phi \in \Phi \tag{5}$$

$$a_s \in \{0, 1\} \qquad \forall s \in \mathcal{S}^P \tag{6}$$

$$y_{s,d}^{t,\phi}, z_{s,d}^{t,\phi} \in \mathbb{R}^+ \qquad \forall s \in \mathcal{S}, d \in \mathcal{D}, t \in \mathcal{T}, \phi \in \Phi. \tag{7}$$

The objective function (1) minimizes the overall costs given by the activation of physical CDN nodes as well as the usage of the virtual infrastructure. In particular, the virtual cost component is computed as the expected value for all the considered scenarios. Constraints (2) set a capacity bound on the overall demand served by physical CDN surrogates. If a physical surrogate is not activated, that is $a_s = 0$, it will not be capable to serve any request. Similarly, the virtual CDN nodes capacity is fixed in (3). In (4) we make sure that the overall clients' demands are served in any time slot and scenario, by virtual or physical surrogate servers. Flows can be split across multiple CDN servers. In (5) we control the overall service quality. We make sure that a fraction of at least ϵ requests in each time slot is served by CDN surrogates within a maximum delay of Δ. Finally, binary restrictions on the activation variables are set in (6), while non-negativity constraints on the continuous flow variables are enforced in (7). Rather than considering the worst or mean case, the stochastic formulation ensures that constraints hold in every scenario, while the objective function is optimized given the uncertainty on future traffic requests.

In order to solve the optimization problem (1)-(7) we employ different strategies:

1. A mixed integer linear programming solver (MILP);
2. The L-shaped algorithm (*single* and *multicut* versions);
3. A polynomial-time greedy heuristic.

In Sec. 3.3 we present the L-shaped and greedy algorithms.

3.3. Stochastic CDN Planning Algorithms

Finding the optimal solution for the stochastic CDN planning problem is extremely time consuming, especially in large-scale, real network scenarios, as those used in our numerical evaluation. Motivated by this observation, we now present two algorithms to efficiently solve the stochastic CDN planning problem: i) the L-shaped algorithm and ii) the greedy algorithm.

3.3.1. L-Shaped Algorithm

The L-shaped algorithm can be used to find an exact solution of the optimization problem we formulated in Sec. 3.2. Theoretical results guarantee that the L-shaped algorithm converges to the optimal solution, but in some cases the speed of convergence might be too slow for the considered application.

The L-shaped algorithm, whose pseudo code is given in Algorithm 1, proceeds according to the following steps (we refer to Birge and Louveaux [6] for an introduction to this algorithm):

- *Step 1*: We initialize indices r, s, ν to zero ($r = s = \nu = 0$) and we define the optimization problem in canonical form:

$$\min \quad \mathbf{c}^T\mathbf{a} + \sum_{\phi\in\Phi} p_\phi \mathbf{q}_\phi^T \mathbf{y}_\phi \tag{8}$$

subject to:

$$T_\phi\mathbf{a} + W\mathbf{y}_\phi = \mathbf{h}_\phi \qquad \phi \in \Phi \tag{9}$$

$$a_s \in \{0, 1\} \tag{10}$$

$$\mathbf{y}_\phi \geq 0 \qquad \phi \in \Phi \tag{11}$$

where $\mathbf{c}$ is the vector of physical CDN server installation costs, $\mathbf{c} = \{\mathcal{C}_1^P, \ldots, \mathcal{C}_{|\mathcal{S}^P|}^P\}$, and $\mathbf{q}_\phi$ is a three-dimensional matrix, which can be straightforwardly expressed in terms of the usage cost of virtual CDN servers. Similarly, when the problem in (1)-(7) is represented in canonical form, the constraints are represented in matricial form, and therefore T_ϕ and $\mathbf{h}_\phi$ can be easily defined as a function of parameters $\mathcal{K}_s^P$, $\mathcal{K}_s^V$, $r_d^{t,\phi}$ and ϵ.

We have two sets of variables: i) *"here-and-now"* variables which are the binary activation variables a_s for the physical CDN nodes and ii) *"wait-and-see"* variables which are the continuous flow variables $\mathbf{y}_\phi$, composed of elements $y_{s,d}^{t,\phi}$ and $z_{s,d}^{t,\phi}$. It is worth noting that it is easier to determine the flow variables $\mathbf{y}_\phi$ once given the nodes activation variables a_s.

- *Step 2*: We increment ν by 1 ($\nu = \nu + 1$) and we solve the Master Problem (MP) (12)-(16):

$$\min \quad \mathbf{c}^T\mathbf{a} + \theta \tag{12}$$

subject to:

$$D_l\mathbf{a} \geq d_l \qquad l = 1, ..., r \tag{13}$$

$$E_l\mathbf{a} + \theta \geq e_l \qquad l = 1, ..., s \tag{14}$$

$$\mathbf{a} \in \{0, 1\} \tag{15}$$

$$\theta \in \mathbb{R}^+ \tag{16}$$

In MP, we consider only the physical nodes activation variables (the MP problem does not contain the stochastic scenarios). The set of inequalities in (13) and (14) represent the feasibility and optimality cuts of MP, respectively. In the first round, these sets are empty. Let ($\mathbf{a}^\nu$, θ^ν) be an optimal solution of MP. If no constraint (14) is present, θ^ν is set equal to $-\infty$ and is not considered in the computation of the vector $\mathbf{a}^\nu$.

- *Step 3*: We assume in this step that the *here-and-now* variables are known (i.e., the vector $\mathbf{a}^\nu$ is computed in the previous step) and we solve, for each

Algorithm 1: L-Shaped Algorithm

```
  Input  : $\langle \mathcal{D}, \mathcal{S}, \mathcal{T}, \Phi, r_d^{t,\phi}, \mathcal{K}_s^P, \mathcal{K}_s^V, \mathcal{C}_s^P, \mathcal{C}_s^V, \delta_{s,d}, \Delta, \epsilon, p_\phi \rangle$
  Output: $\hat{a}, \hat{y}_{s,d}^{t,\phi}, \hat{z}_{s,d}^{t,\phi}$, min_cost
1 Initialization: $r = s = \nu = 0$;
  represent the optimization problem (1)-(7) in canonical form: (8)-(11);
2 $\nu = \nu + 1$;
  solve the Master Problem (MP) defined in (12)-(16);
  $(\mathbf{a}^\nu, \theta^\nu)$ = get_MP_solution();
3 foreach $\phi \in$ Scenarios do
      solve the Linear Problem (LP) in (17)-(19);
      $w'$ = get_LP_solution_ObjFunValue();
      if $w' \geq 0$ then
          compute $D_{r+1}$ and $d_{r+1}$ according to (20) and (21);
          generate a feasibility cut; $r = r + 1$;
          add the feasibility cut to the MP problem; go to Step 2;
      end
  end
4 foreach $\phi \in$ Scenarios do
      solve the LP in (22)-(24);
      $(\hat{y}_{s,d}^{t,\phi}, \hat{z}_{s,d}^{t,\phi})$ = get_LP_solution($a^\nu$);
      compute $E_{s+1}$ and $e_{s+1}$ according to (25) and (26);
  end
  Let $w^\nu = e_{s+1} - E_{s+1} a^\nu$;
  if $\theta^\nu < w^\nu$ then
      generate an optimality cut; $s = s + 1$;
      add the optimality cut to the MP problem; go to Step 2;
  end
  Optimal solution found: $\hat{a} = a^\nu$;
  min_cost = calculate_ObjFunValue($\hat{a}$, $\hat{y}_{s,d}^{t,\phi}$, $\hat{z}_{s,d}^{t,\phi}$);
  return($\hat{a}$, $\hat{y}_{s,d}^{t,\phi}$, $\hat{z}_{s,d}^{t,\phi}$, min_cost);
```

scenario $\phi \in \Phi$, the following linear problem:

$$\min \quad w' = (1, \ldots, 1)\mathbf{v}^{+} + (1, \ldots, 1)\mathbf{v}^{-} \tag{17}$$

subject to:

$$W\mathbf{y} + I\mathbf{v}^{+} - I\mathbf{v}^{-} = \mathbf{h}_{\phi} - T_{\phi}\mathbf{a}^{\nu} \tag{18}$$

$$\mathbf{y} \geq 0, \mathbf{v}^{+} \geq 0, \mathbf{v}^{-} \geq 0 \tag{19}$$

If for some ϕ the optimal value $w' > 0$, then let σ^{ν} be the corresponding multiplier and we define D_{r+1} and d_{r+1} as follows:

$$D_{r+1} = (\sigma^{\nu})^{T} T_{\phi} \tag{20}$$

$$d_{r+1} = (\sigma^{\nu})^{T} h_{\phi} \tag{21}$$

D_{r+1} and d_{r+1} are used to generate a constraint called a *feasibility cut* of type (13). Then, r is incremented by 1 and this feasibility cut is added to the set (13). At this point, go to Step 2. When for all $\phi \in \Phi$ $w' = 0$, go to Step 4.

- *Step 4*: As in Step 3, for each scenario $\phi \in \Phi$, we solve the following linear problem:

$$\min \quad \mathbf{q}_{\phi}^{T}\mathbf{y} \tag{22}$$

subject to:

$$W\mathbf{y} = \mathbf{h}_{\phi} - T_{\phi}\mathbf{a}^{\nu} \tag{23}$$

$$\mathbf{y} \geq 0 \tag{24}$$

Then we define the multiplier π_{ϕ}^{ν} associated with the optimal solution of the above problem (for each $\phi \in \Phi$). In this step, we define E_{s+1} and e_{s+1} as follows:

$$E_{s+1} = \sum_{\phi \in \Phi} p_{\phi} (\pi_{\phi}^{\nu})^{T} T_{\phi} \tag{25}$$

$$e_{s+1} = \sum_{\phi \in \Phi} p_{\phi} (\pi_{\phi}^{\nu})^{T} h_{\phi} \tag{26}$$

Let $w^\nu = e_{s+1} - E_{s+1} a^\nu$. If $\theta^\nu \geq w^\nu$, stop; a^ν is an optimal solution. Otherwise, $s = s+1$ and we generate the constraints (called the *optimality cuts*) using the above calculated terms in (25) and (26), and we add the *optimality cuts* to the set (14) and then the algorithm returns to Step 2.

3.3.2. Greedy Algorithm

In this section we describe the polynomial-time heuristic we designed to compute a close-to-optimal solution to the planning problem. The pseudo-code is provided in Algorithm 2 to describe the steps we use to achieve this purpose.

At the beginning of the algorithm, in Step 1, all the physical nodes are activated and sorted. In particular, we give higher priority to those nodes that can serve the largest amount of traffic demands within a delay of Δ. Given the

Algorithm 2: Greedy Algorithm

```
  Input  : $\langle \mathcal{D}, \mathcal{S}, \mathcal{T}, \Phi, r_d^{t,\phi}, \mathcal{K}_s^P, \mathcal{K}_s^V, \mathcal{C}_s^P, \mathcal{C}_s^V, \delta_{s,d}, \Delta, \epsilon, p_\phi \rangle$
  Output: $\hat{a}, \hat{y}_{s,d}^{t,\phi}, \hat{z}_{s,d}^{t,\phi}$, min_cost
1 $\hat{a} = [1; 1; ...; 1]$; phy_nodes = sort_phy_CDN_nodes();
2 if $\neg$is_feasible($\hat{a}$) then
      return INFEASIBLE_ASSIGNMENT ;
  end
  $\langle \hat{y}_{s,d}^{t,\phi}, \hat{z}_{s,d}^{t,\phi}$, min_cost$\rangle$ = get_LP_solution($\hat{a}$); best_sol = $\hat{a}$;
3 foreach $s \in$ phy_nodes do
      $\hat{a}_s = 0$ ;
      $\langle \hat{y}_{s,d}^{t,\phi}, \hat{z}_{s,d}^{t,\phi}$, current_min_cost$\rangle$ = get_LP_solution($\hat{a}$);
      if current_min_cost $\geq$ min_cost then
          break ;
      end
      min_cost = current_min_cost ; best_sol = $\hat{a}$;
  end
4 $\hat{a}$ = best_sol ; $\langle \hat{y}_{s,d}^{t,\phi}, \hat{z}_{s,d}^{t,\phi}$, min_cost$\rangle$ = get_LP_solution($\hat{a}$);
```

structure of the problem, infeasible solutions are those that cannot be served even when all the physical CDN nodes have been activated.

In Step 2 we check this condition and eventually signal a potential infeasibility. The *get_LP_solution* function computes a solution for the continuous relaxation of the model (1)-(7), using a standard linear programming solver, and therefore has a polynomial-time complexity. As outputs, it returns the optimal flows for the physical and virtual CDN nodes as well as the overall cost. In case of infeasibility, the output value of *min_cost* is set to infinite.

The loop in Step 3 deactivates at every iteration one new physical node, according to the previously generated ordering, and it completes when the objective function does not improve anymore. Lastly, in Step 4 we compute the optimal flows starting from the best physical nodes allocation choice.

4. Numerical Results

In this section we present the numerical results we obtained performing a thorough analysis of our models and heuristics under realistic network conditions.

Unless stated otherwise, our network topology is created using the Barabási–Albert model and is composed of 50 consumers, 20 physical and 15 virtual CDN nodes. Traffic demands are generated using as a reference the Cisco VNI data for the 2014-2017 years: the planning horizon is of 3 years and we used 36 different time slots. Traffic uncertainty is taken into account by considering 10 different traffic scenarios, with a variable overall demand between 80% and 120% of the Cisco forecast. To control the overall demand, we limit to 20 Gbit/s the maximum traffic requests that a consumer can generate in a time slot of a scenario. Physical CDN nodes have a capacity of 12.5 Gbit/s, while virtual CDNs can serve up to 8 Gbit/s. Similar trends have been observed for other capacity values, omitted here for the sake of brevity. Link delays are generated in the range of those available on Rocketfuel for the Sprintlink (US) topology, with an average delay of 3 ms. Moreover, we assume the CDN provider wants to guarantee that at least 95% of the requests are served by surrogates with a delay lower than 12 ms, that is, selected CDN nodes are, on average, 4 hops far from the consumer's location.

Lastly, prices are set as follows: we assume that the cost to install and operate one CDN node is set in the range [8; 12] kUSD, while different prices will be considered for virtual CDN nodes in the range [0.001; 10] USD per Mbit/s. For the same physical-virtual price ratios, even by considering different values for the physical CDN pricing, we observed similar trends as those discussed in this section. Hereafter, we report the result we obtained using CPLEX 12.5 [25] as a MILP solver, bounding the maximum execution time of the algorithms to 1 hour (with a 5% MIP gap), and using a machine equipped with a quad-core Intel i7-3770 (3.40 GHz) CPU with 16 Gbyte of RAM. Lastly, for each of the results we performed 20 different runs and we report the narrow 95% confidence intervals.

We first present an example network scenario (Sec. 4.1), then we analyze NFV benefits in large-scale topologies in terms of network cost, studying the impact of the physical CDN capacity, the number of clients, the number of physical and virtual nodes, traffic uncertainties and minimum service level to be guaranteed (Sec. 4.2). The effect of vCDN pricing is then discussed (Sec. 4.3), along with the computing time necessary to obtain the solution (Sec. 4.4).

4.1. Example network scenario.

To highlight the impact of the Δ and ϵ parameters (the maximum tolerated delay and the minimum service level guaranteed, respectively), expressed in constraints (5), on the solution of the optimization model (1)-(7), we present here an example scenario, where 5 physical CDNs (with a capacity of 50 Gbit/s each) are deployed.

Fig. 2 shows the objective function value (the overall cost) as a function of these two parameters. In particular, the lower the tolerated delay Δ, the higher the overall cost the operator must incur to ensure the feasibility of the solution. In fact, the overall cost for $\Delta = 25$ ms is up to ≈ 5 times higher than that

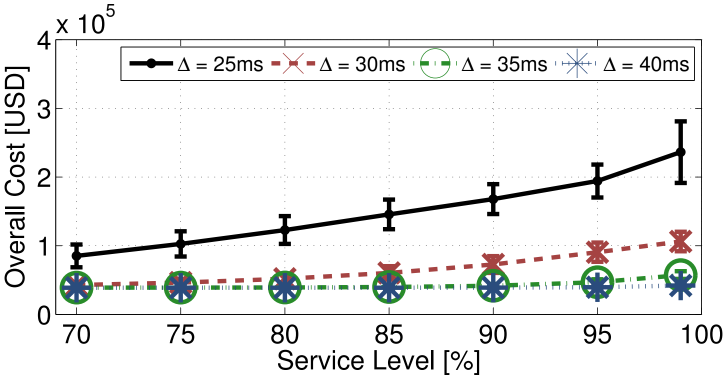
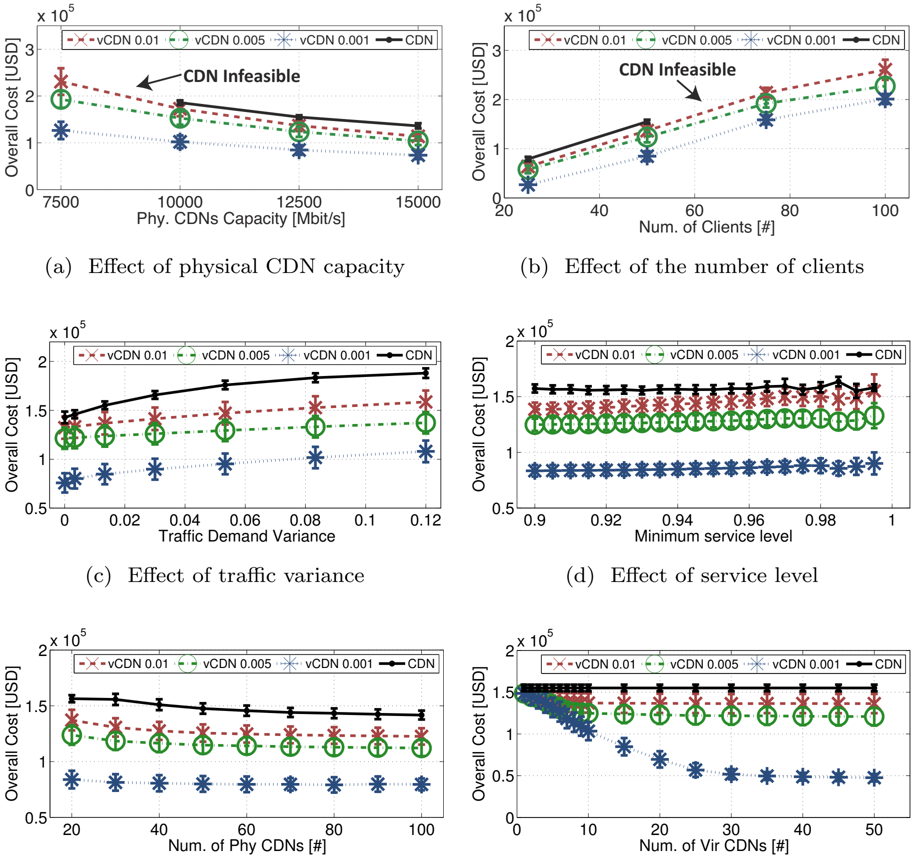


Figure 2: *Effect of the Service Level and Delay.* The figure shows the overall cost as a function of the desired service level (ϵ) and for different delays ($\Delta = \{25, 30, 35, 40\}$ ms).

obtained with $\Delta = 40$ ms. Similarly, the higher the value of ϵ, the higher the overall cost, which more than doubles when the required service level increases from 70% to 95% for $\Delta = 35$ and 40 ms, for example. In particular, this behavior is explained by the fact that the operator must use in these cases virtual CDN nodes to serve the traffic requests within the maximum tolerated delay, while guaranteeing the minimum service level. On the other hand, when Δ is large (or ϵ is small), the model can leverage the services provided by physical CDNs even though they are far from the content consumers, and thus decreases the overall operator cost.

4.2. NFV Benefits

Fig. 3 shows the cost benefits for the mixed physical-virtual CDN architecture, considering different virtual prices. As expected, in Fig. 3a-3f, lower CDN prices lead to lower overall costs. The effect of the *physical CDN capacity* is shown in Fig. 3a. If the capacity of the physical CDN appliances is lower than 10 Gbit/s, using virtual nodes becomes mandatory since otherwise an infeasibility is produced. When the physical CDN nodes capacity is set to 15 Gbit/s, cost savings up to 46% are experienced for cheap virtual CDN pricing (i.e, 0.001 USD per unit of bandwidth), whereas the saving is reduced to 16% if we set a vCDN price of one order of magnitude larger (i.e, 0.01 USD per unit of bandwidth).



(a) Effect of physical CDN capacity

(b) Effect of the number of clients

(c) Effect of traffic variance

(d) Effect of service level

(e) Effect of the number of physical CDNs

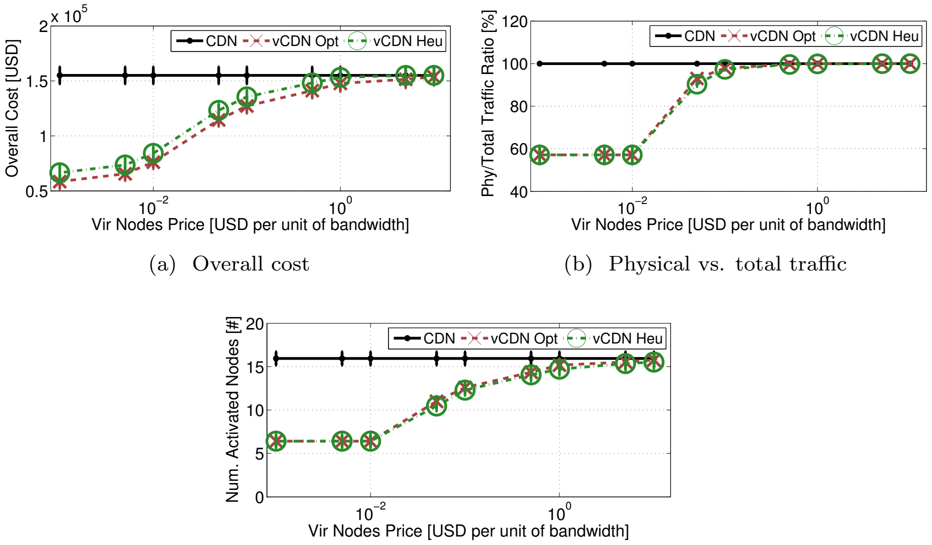
(f) Effect of the number of virtual CDNs

Figure 3: *NFV Benefits.* Plots 3a-3f show the overall cost benefits of an architecture composed of a mix of physical and virtual CDN nodes (denoted in the figures with vCDN), with respect to the scenario where only the physical CDN infrastructure is used (denoted in the figures with CDN). Different prices are considered for the vCDN case, as shown in the legend (0.01, 0.005, 0.001). The impact of several parameters is investigated, including the physical CDN capacity, the number of clients, the traffic variance, the desired service level, the number of available physical and virtual nodes.

Fig. 3b shows the effect of the *number of clients* on the costs. The physical-only CDN infrastructure cannot handle more than 50 clients, whereas up to 100 clients can be served if we also leverage the 15 virtual CDNs deployed. The effects of *traffic uncertainties* are quantified in Fig. 3c, where we show the overall cost as a function of the traffic demand variance (starting from 0, i.e. from perfect traffic knowledge, and increasing it). With the largest variance that we took into account, the vCDN infrastructure leads to cost savings in the range 16-43% according to the vCDN pricing.

Fig. 3d shows the effect of the *minimum service level* (ϵ) on the cost. It can be seen that in the four considered cases the cost increases with ϵ. Note that the impact of ϵ on the solution of the stochastic CDN planning problem can be more appreciated when the number of physical and virtual nodes is smaller, and to guarantee feasibility, virtual CDN nodes must be used more extensively.

Finally, in Fig. 3e and Fig. 3f we plot the total cost the operator incurs when varying the *number of physical* and *virtual CDN nodes*, respectively. As expected, the higher the number of physical nodes deployed in the network, the lower is the cost for the operator; this trend is the same for the four considered cases. Given that the number of physical CDN servers is fixed and equal to 20, in Fig. 3f we plot the overall cost obtained when the number of virtual nodes increases from 1 to 50. We can observe from this figure that for a small-to-medium number of virtual CDN nodes (i.e., $\leq$30), the planning cost significantly decreases with this parameter (from 1.5 10^5 to 0.5 10^5 USD, especially, when the cost per unit of bandwidth is 0.001 (the "vCDN 0.001" case) and then it stabilizes (at 0.5 10^5 USD) for large values. The trend in Fig. 3e and Fig. 3f is due to the fact that a medium number of virtual and physical nodes is sufficient to serve all the traffic requests of consumers while guaranteeing a minimum service level of 95%; we intuitively expect that for a small number of physical CDN nodes, more and more virtual nodes are needed in order to serve consumers at higher service levels.



(a) Overall cost

(b) Physical vs. total traffic

(c) Physical nodes activation

Figure 4: *Effect of the vCDN Price.* Plots 4a-4c show the effect of the virtual nodes pricing on the overall cost (Fig. 4a), the fraction of traffic served by the physical infrastructure (Fig. 4b) and the number of physical nodes activated (Fig. 4c), for different solution algorithms as well as a physical-only infrastructure (denoted in the figures with CDN).

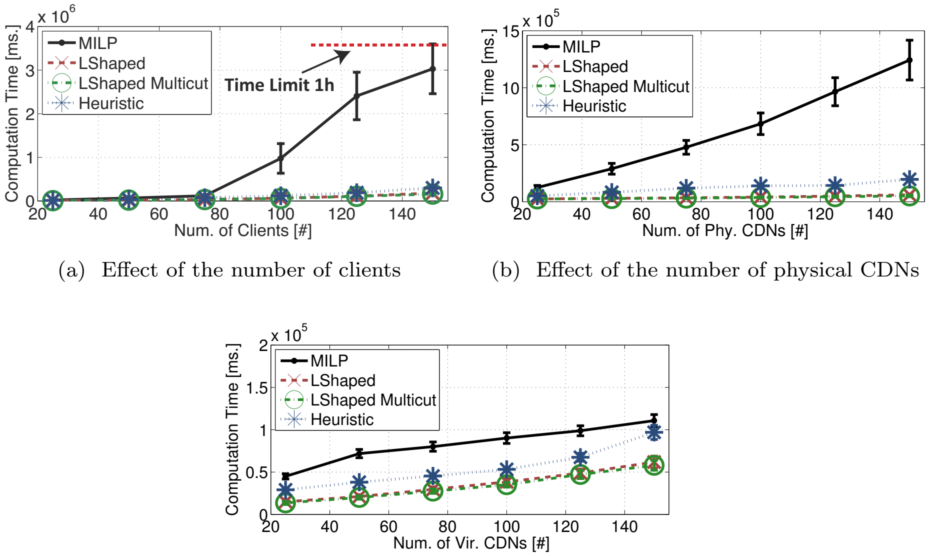
4.3. Effect of the vCDN Price

Due to its remarkable effect, in Fig. 4 we show the impact of the virtual CDN pricing using the different solution algorithms considered in this work. The overall cost is portrayed in Fig. 4a. Solutions obtained with exact solvers such as the deterministic equivalent program in the extensive form or the L-shaped algorithm lead to costs up to 11% (and on average 6%) lower than those reported with the greedy heuristic. Fig. 4a clearly shows that the higher the prices, the lower the economic benefits of using a mixed physical-virtual CDN infrastructure. In particular, considering prices in the range $[0.001; 0.5]$ USD per unit of bandwidth, the cost savings compared to the physical-only solution are in the range 5-64%. Fig 4b shows the proportion of physical traffic with respect to the overall demand, as a function of the price of virtual nodes. Cheap prices make the virtual CDN capacity be fully saturated, and for this reason the left hand-side of Fig. 4b has an horizontal trend that accounts for 57% of the overall traffic. As a consequence, Fig. 4c shows that the number of activated physical CDN nodes does not increase for virtual prices lower than 0.01 USD. Comparing Fig. 4b and 4c, the slope of the curve is less steep in the second plot, since there are cases where it is convenient to strategically deploy a physical CDN server in a special position of the topology even though it is not fully used by the clients. It is interesting to note that for both Fig. 4b and 4c the heuristic algorithm leads to solutions that are practically overlapped to the optimal choice.

4.4. Computing time

To limit the effects of infeasibilities that negatively affect results on the computing time, hereafter we raise the number of virtual CDN nodes to 50, making the network be capable to serve up to 150 consumers. Fig. 5 shows the execution time of the different algorithms as a function of the number of clients (Fig. 5a), the number of physical (Fig. 5b), and virtual CDN nodes (Fig. 5c).

The number of clients is the parameter that mostly affects the execution time, as portrayed in Fig. 5a. In particular, the MILP solver for the deterministic equivalent program has a time trend that is exponential in the number



(a) Effect of the number of clients

(b) Effect of the number of physical CDNs

(c) Effect of the number of virtual CDNs

Figure 5: *Execution time.* Plots 5a-5c show the behavior of the different solution algorithms as a function of the number of clients as well as the number of surrogate nodes. We observe that the number of clients has the most remarkable effect on the execution time.

of served clients (Fig. 5a), but linear in the number of physical (Fig. 5b) and virtual CDNs (Fig. 5c). While the MILP solver can hardly scale to topologies with a larger number of nodes, this possibility is instead offered by the L-shaped decomposition and our proposed heuristic. As a matter of fact, all these algorithms can solve the planning problem saving up to 94% of time compared to the MILP problem, when considering 150 clients, as shown in Fig. 5a. Lastly, although there are cases where the heuristic algorithm is slightly slower than the L-shaped algorithm (as in Fig. 5c), we remark the fact that the heuristic has a worst-case polynomial time complexity, whereas a comparable theoretical result for the L-shaped algorithm does not hold.

5. Conclusion

In this paper we tackled the stochastic planning problem for content delivery to study potential benefits that Network Functions Virtualization can provide for content distribution purposes. We considered a mixed architecture where both physical as well as virtual CDN nodes can be used by a CDN owner to implement the content distribution service. The owner performs the planning choice for the physical CDN infrastructure on a long-term time schedule, possessing only a stochastic estimate of future traffic demands.

Our study shows that a mixed solution where both virtual an physical CDN nodes are used can dramatically reduce the overall costs sustained by the operator to purchase and operate the distribution infrastructure. In particular, we observed that gains can be up to 65% when considering the cheapest vCDN price. Our contribution is also to formulate efficient solution algorithms for the two-stage stochastic planning problem that can scale to realistic topology sizes. Rather than solving the deterministic equivalent problem in the extensive form, our proposed L-shaped algorithm and the greedy heuristic can efficiently find a solution, saving up to 95% of time compared to the MILP solver.

References

[1] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, Z.-L. Zhang, Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery, in: Proc. of IEEE INFOCOM, IEEE, Orlando, Florida, USA, Mar. 2012, pp. 1620–1628. doi:10.1109/INFCOM.2012.6195531.

[2] H. H. Liu, Y. Wang, Y. R. Yang, H. Wang, C. Tian, Optimizing Cost and Performance for Content Multihoming, in: Proc. of ACM SIGCOMM, Vol. 42, ACM, Helsinki, Finland, Aug. 2012, pp. 371–382. doi:10.1145/2377677.2377753.

[3] E. Nygren, R. K. Sitaraman, J. Sun, The Akamai Network: A Platform for High-Performance Internet Applications, ACM SIGOPS Operating Systems Review 44 (3) (July 2010) 2–19. doi:10.1145/1842733.1842736.

[4] B. Niven-Jenkins, F. L. Faucheur, N. Bitar, Content Distribution Network Interconnection (CDNI) Problem Statement, RFC 6707 (Informational) (Sep. 2012).

[5] ETSI, Network Functions Virtualisation - White Paper v.3, https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf, Last accessed: Apr. 2016.

[6] J. R. Birge, F. Louveaux, Introduction to Stochastic Programming, 2nd Edition, Springer Series in Operations Research and Financial Engineering, New York, NY, 2011. doi:10.1007/978-1-4614-0237-4.

[7] J. Matias, J. Garay, N. Toledo, J. Unzilla, E. Jacob, Toward an SDN-enabled NFV architecture, IEEE Communications Magazine 53 (4) (April 2015) 187–193. doi:10.1109/MCOM.2015.7081093.

[8] A. Khan, A. Zugenmaier, D. Jurca, W. Kellerer, Network virtualization: a hypervisor for the Internet?, IEEE Communications Magazine 50 (1) (Jan 2012) 136–143. doi:10.1109/MCOM.2012.6122544.

[9] Q. Duan, Y. Yan, A. Vasilakos, A Survey on Service-Oriented Network Virtualization Toward Convergence of Networking and Cloud Computing, IEEE Transactions on Network and Service Management 9 (4) (Dec 2012) 373–392. doi:10.1109/TNSM.2012.113012.120310.

[10] G. Roy, S. Saurabh, N. Upadhyay, P. Gupta, Creation of virtual node, virtual link and managing them in network virtualization, in: Proc. of 2011 World Congress on Information and Communication Technologies (WICT), IEEE, Mumbai, India, Dec. 2011, pp. 738–742. doi:10.1109/WICT.2011.6141338.

[11] X. Costa-Perez, J. Swetina, T. Guo, R. Mahindra, S. Rangarajan, Radio access network virtualization for future mobile carrier networks., IEEE Communications Magazine 51 (7) (July 2013) 27–35. doi:10.1109/MCOM.2013.6553675.

[12] A. Manzalini, et al., Software-defined networks for future networks and services. main technical challenges and business implications, in: Proc. of IEEE Workshop Software Defined Networks for Future Networks and Services (SDN4FNS), IEEE, 2014.

[13] N. M. K. Chowdhury, R. Boutaba, Network Virtualization: State of the Art and Research Challenges, IEEE Communications Magazine 47 (7) (2009) 20–26. doi:10.1109/MCOM.2009.5183468.

[14] A. Fischer, J. Botero, M. Till Beck, H. de Meer, X. Hesselbach, Virtual Network Embedding: A Survey, IEEE Communications Surveys Tutorials 15 (4) (2013) 1888–1906. doi:10.1109/SURV.2013.013013.00155.

[15] A. Jarray, A. Karmouch, Decomposition Approaches for Virtual Network Embedding With One-Shot Node and Link Mapping, IEEE/ACM Transactions on Networking 23 (3) (June 2015) 1012–1025. doi:10.1109/TNET.2014.2312928.

[16] N. M. K. Chowdhury, M. R. Rahman, R. Boutaba, Virtual network embedding with coordinated node and link mapping, in: Proc. of IEEE INFOCOM, IEEE, Rio de Janeiro, Brazil, Apr. 2009, pp. 783–791. doi:10.1109/INFCOM.2009.5061987.

[17] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, J. Wang, Virtual Network Embedding Through Topology-Aware Node Ranking, ACM SIGCOMM Computer Communication Review 41 (2) (2011) 38–47. doi:10.1145/1971162.1971168.

[18] X. Guan, B.-Y. Choi, S. Song, Energy efficient virtual network embedding for green data centers using data center topology and future migration, Computer Communications 000 (2015) 1–10. doi:10.1016/j.comcom.2015.05.003.

[19] J. M. Wang, J. Zhang, B. Brahim, Content Multi-homing: an Alternative Approach, in: Proc. of the IEEE Int.l Conf. on Communications (ICC), IEEE, Sydney, Australia, Jun. 2014, pp. 371–382. doi:10.1109/ICC.2014.6883800.

[20] J. Yao, H. Zhou, J. Luo, X. Liu, H. Guan, COMIC: Cost Optimization for Internet Content Multihoming, IEEE Transactions on Parallel and Distributed Systems 26 (7) (July 2015) 1851–1860. doi:10.1109/TPDS.2014.2330589.

[21] A. Atamtürk, M. Zhang, Two-Stage Robust Network Flow and Design Under Demand Uncertainty, Operations Research 55 (4) (2007) 662–673. doi:10.1287/opre.1070.0428.

[22] X. Liu, The Role of Stochastic Programming in Communication Network Design, Computers & Operations Research 32 (9) (2005) 2329–2349. doi:10.1016/j.cor.2004.03.006.

[23] A. Eisenblätter, J. Schweiger, Multistage Stochastic Programming in

Strategic Telecommunication Network Planning, Computational Management Science 9 (3) (2012) 303–321. doi:10.1007/s10287-012-0143-5.

[24] F. Figueiredo, F. Benevenuto, J. M. Almeida, The Tube over Time: Characterizing Popularity Growth of YouTube Videos, in: Proc. of the 4th Int.l Conf. on Web Search and Data Mining (WSDM), ACM, Hong Kong, Feb. 2011, pp. 745–754. doi:10.1145/1935826.1935925.

[25] IBM ILOG CPLEX Optimizer, version 12.5 (2015).