

Virtual Network Functions Placement and Routing Optimization

Bernardetta Addis*, Dallal Belabed[†], Mathieu Bouet[‡], Stefano Secci[†]

*LORIA, France. Email: bernardetta.addis@loria.fr.

[†] Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6. Email: {dallal.belabed, stefano.secci}@upmc.fr

[‡]Thales Communications & Security. Email: mathieu.bouet@thalesgroup.com

Abstract—Network Functions Virtualization (NFV) is incrementally deployed by Internet Service Providers (ISPs) in their carrier networks, by means of Virtual Network Function (VNF) chains, to address customers' demands. The motivation is the increasing manageability, reliability and performance of NFV systems, the gains in energy and space granted by virtualization, at a cost that becomes competitive with respect to legacy physical network function nodes. From a network optimization perspective, the routing of VNF chains across a carrier network implies key novelties making the VNF chain routing problem unique with respect to the state of the art: the bitrate of each demand flow can change along a VNF chain, the VNF processing latency and computing load can be a function of the demands traffic, VNFs can be shared among demands, etc. In this paper, we provide an NFV network model suitable for ISP operations. We define the generic VNF chain routing optimization problem and devise a mixed integer linear programming formulation. By extensive simulation on realistic ISP topologies, we draw conclusions on the trade-offs achievable between legacy Traffic Engineering (TE) ISP goals and novel combined TE-NFV goals.

I. INTRODUCTION

With the emergence of Network Functions Virtualization (NFV) [1], [2] the attention of network virtualization research is now focusing on key aspects of NFV systems that were either not considered relevant or not conceived before industry effort at Standards Developing Organizations (SDOs). Key aspects that are worth being mentioned are the:

- NFV service chaining provisioning;
- flow orchestration over VNF chains as a function of demand assignment to existing VNF chains or sub-chains;
- ingress/egress bit-rate variations at VNFs due to specific VNF operations (e.g., compression/decompression);
- VNF processing and forwarding latency as an orchestration parameter to take into account for emerging fastpath solutions such as [3].

ETSI is de-facto the reference SDO for the NFV high-level functional architecture specification. ETSI specifies three layers for the NFV architecture: Virtual Network Functions (VNFs), the nodes; NFV Infrastructure (NFVI), including the elements needed to run VNFs such as the hypervisor node and the virtualization clusters; MANagement and Orchestration (MANO), handling the operations needed to run, migrate, optimize VNF nodes and chains, possibly in relationship with transport network orchestrators.

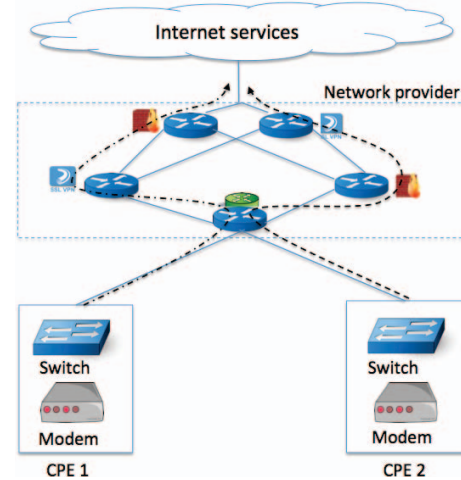


Fig. 1. VNF chaining with virtualized Customer Premises Equipment (vCPE).

A promising NFV use-case for carrier networks is the virtual Customer Premises Equipment (vCPE) that simplifies the CPE equipment by means of virtualized individual network functions placed at access and aggregation network locations, as depicted in Fig. 1. MANO operations must take into consideration the special nature of NFV architectures, such as the latency/traffic bounds at both the VNF node and the end-to-end levels, the fact that some VNFs can modify the incoming bitrate by compressing or decompressing it, etc. In this context, the paper contribution is as follows:

- we define and formulate via mathematical programming the VNF Placement and Routing (VNF-PR) optimization problem, including compression/decompression constraints and two forwarding latency regimes (with and without fastpath), under both TE and NFV objectives.
- we design a math-heuristic approach allowing us to run experiments also for large instances of the problem within an acceptable execution time.
- we evaluate our solution on realistic settings. We draw considerations on NFV deployment strategies.

The paper is organized as follows. Section II presents the state of the art on NFV orchestration. Section III describes the network model and the Mixed Integer Linear Programming (MILP) formulation. Analysis and discussion of optimization results are given in Section IV. Section V concludes the paper.

II. BACKGROUND

In the state of the art, preliminary works on NFV orchestration tend to map the problem into a Virtual Network Embedding (VNE) problem. This is for example the case of [4], where VNFs are treated as normal virtual machines (VMs) to be mapped on a network of VM containers interconnected via physical links that host logical links of virtual network demands. Similarly, authors in [5] propose a VNF chaining placement that combines location-routing and VNE problems, solving first the placement and then the chaining. In [6] the authors decouple the legacy VNE problem into two embedding problems: VNF chaining and VM embeddings, where a service chain is embedded on a VM, and each VM on physical servers. Each service chain has specific requirements as notably an end-to-end latency requirement.

The placement and routing of VNF chains is a problem fundamentally different from the VNE problem. As in VNE, virtual network nodes need to be placed in an underlying physical infrastructure. However, differently from VNE, in VNF chaining: (i) the demand is not a multipoint-to-multipoint network connection request, but as a point-to-point source-destination flow routing demand and (ii) specific aspects of NFV such as forwarding latency behavior, ingress/egress bit-rate changes, and chaining are not addressed in VNE, and their inclusion would further increase the time complexity. In this sense VNF chaining problem is closer to facility location problems, whereas VNE is a mapping problem.

We believe the appropriate way to deal with NFV MANO decision problems is to define the VNF Placement and Routing (VNF-PR) problem directly tailored to the NFV environment, for the sake of time complexity, modeling precision and practical usability. This is also the approach adopted by a few papers in the literature [7], [8], [9]. In [7] the authors consider the online orchestration of VNFs, modeling it as a scheduling problem of VNFs and proposing heuristics to scale with the online nature of the framework. In [8] the authors consider a VNF-PR problem for data-center environments with both optical and electronic elements, formulating the problem as a binary integer programming problem, and propose a heuristic algorithm to solve it. In their work, VNF chains are set as an input to the problem. In [9] the specific Deep Packet Inspection (DPI) VNF node placement problem (with no chaining) is targeted, with a formal definition of the problem and a greedy heuristic algorithm to solve it. Our paper takes inspiration from these early works, yet goes beyond being more generic and integrating the specific features of NFV environments mentioned in the introduction.

III. NETWORK MODEL

We provide in the following a definition of the VNF Placement and Routing problem, and its formulation.

A. Problem statement

Definition Virtual Network Function Placement and Routing (VNF-PR) Problem

Given a network graph $G(N, A)$, where N is the set of

nodes, A the set of arcs between nodes, $N_v \subseteq N$ the set of nodes disposing of NFVI server clusters. Given a set of edge demands D , each demand $k \in D$ being characterized by a source o_k , a destination t_k , a nominal bandwidth b_k statistically representative for demand k , and a set of VNFs of different types to be traversed by edge flows, the VNF-PR optimization problem is to find:

- the optimal placement of VNF nodes over NFVI clusters;
- the optimal assignment of demands to VNF node chains.

subject to:

- NFVI cluster capacity constraints;
- VNF flow compression/decompression constraints;
- VNF forwarding latency constraints;
- VNF node sharing constraints.

The optimization objective should contain both network-level and NFVI-level performance metrics. In our network model, we propose as network-level metric a classical TE metric, i.e., the minimization of the maximum link utilization. As NFVI-level metric we propose the minimization of allocated computing resources. Furthermore, we assume that:

- Multiple VNFs of the same type (i.e., same functionality) can be allocated on the same node, but each demand cannot split its flow on multiple VNF of the same type.
- There are different VM templates for embedding a VNF, each with a different computing resource consumption and VNF forwarding latency performance.
- The VNF computing resource consumption can be expressed in terms of live memory (e.g., RAM) and Computing Processing Units (CPUs), yet the model shall be versatile enough to integrate other computing resources.
- Latency introduced by a VNF can follow one among the two following regimes (as represented in Fig. 2):
 - *Standard*: VNFs bufferize traffic at input and output virtual and physical interfaces such that the forwarding latency can be considered as a convex piece-wise linear function of the aggregate bit-rate at the VNF, due to increased buffer utilization and packet loss as the bitrate grows as shown in [3], [10]. This is the case of default VNFs functioning with standard kernel and hypervisor buffers and sockets.
 - *Fastpath*: VNFs use optimally dimensioned and relatively small buffers, and decrease the number of times packets are copied in memory, so that the forwarding latency is constant up to a maximum aggregate bit-rate after which packets are dropped; e.g., this happens for fastpath solutions such as [3].

Fig. 2 gives examples of forwarding latency profiles for the two cases (with two templates used for the tests).

- For each demand and NFVI cluster, only one compression/decompression VNF can be installed. This allows us to keep the execution time at acceptable levels, without reducing excessively the VNF placement alternatives.
- Each network node is interconnected with a NFVI cluster node (in the notations, to simplify, we use a single identifier for collocated network and NFVI nodes).

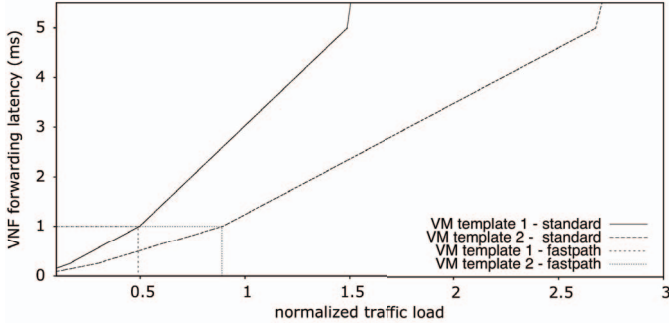


Fig. 2. Example of VNF forwarding latency profiles.

B. Mathematical formulation

Table I reports the mathematical notations used in the following Mixed Integer Linear Programming (MILP) formulation of the VNF-PR problem. We work on an extended graph (to distinguish between origin/destination nodes and NFVI nodes), in which each access node i is duplicated in a node i' . Arc (i, i') will be added and all arcs (i, j) originating from access node i will be transformed in arcs (i', j) .

We use three binary variables: x_{ij}^k represents the per-demand link utilization, hence the path used by the demand; y_{ift}^n represents the allocation of a copy of a VNF on a given node; z_{ift}^{kn} represents the assignment of a given demand to a certain copy of a VNF. To introduce the possibility of compressing/decompressing flows for some VNFs, we need to introduce the explicit flow variable ϕ_{ij}^k and a compression parameter μ_f for each type of VNFs. Furthermore, ψ_{if}^{kn} represents the flow of demand k entering node i and using the copy n of the VNF of type f .

We now present the constraints. Single path flow balance:

$$\sum_{j:(i,j) \in A} x_{ij}^k - \sum_{j:(j,i) \in A} x_{ji}^k = \begin{cases} 1 & \text{if } i = o_k \\ -1 & \text{if } i = t_k \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in D, \forall i \in N \quad (1)$$

Flow and compression/decompression balance:

$$\sum_{j \in N: (i,j) \in A} \phi_{ij}^k - \sum_{j \in N: (j,i) \in A} \phi_{ji}^k = \sum_{f \in F, n \in C_f} (1 - \mu_f) \psi_{if}^{kn} \quad \forall k \in D, \forall i \in N_v \quad (2)$$

Flow balance for access nodes:

$$\begin{aligned} & \sum_{j \in N: (i,j) \in A} \phi_{ij}^k - \sum_{j \in N: (j,i) \in A} \phi_{ji}^k = \\ & = \begin{cases} b^k & \text{if } i = o_k \\ -b^k \cdot \prod_{f \in F: m_{kf}=1} \mu_f & \text{if } i = t_k \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in D, \forall i \in N_a \end{aligned} \quad (3)$$

Coherence between path and flow variables:

$$\phi_{ij}^k \leq b_k^{max} x_{ij}^k \quad \forall k \in D, \forall (i, j) \in A \quad (4)$$

$$\phi_{ij}^k \geq b_k^{min} x_{ij}^k \quad \forall k \in D, \forall (i, j) \in A \quad (5)$$

TABLE I
MATHEMATICAL NOTATIONS

sets	
N_a	access nodes
N'_a	duplication of access nodes, where demands are located
N_c	aggregation/core nodes
N_v	nodes where a VNF can be located, $N_v = N'_a \cup N_c$
N	set of all nodes, $N = N_a \cup N_v$
A	set of all arcs, $A \subseteq N \times N$
D	demands
R	resource types (CPU, RAM, storage)
F	VNF types
C_f	set of possible copies of VNF of type f
T	set of VM template configurations
demand parameters	
o_k	origin of demand $k \in D$
t_k	destination of demand $k \in D$
b_k	nominal bandwidth of demand $k \in D$
b_k^{min}	minimal bandwidth of demand $k \in D$
b_k^{max}	maximal bandwidth of demand $k \in D$
m_{kl}	1 if demand $k \in D$ requests VNF $l \in F$
network parameters	
γ_{ij}	arc capacity
λ_{ij}	link latency
M_i	maximum traffic volume that can be switched
Γ_{ir}	capacity of node $i \in N$ in terms of resource of type $r \in R$
NFV parameters	
rr_{rt}	demand of resource $r \in R$ for a VM of type t
μ_f	compression/decompression factor for VNF $f \in F$
$g_{fj}^t(b)$	j -th latency function of $f \in F$ and aggregate bandwidth b if allocated on VM of type t , linear in requested bandwidth
L	maximum allowed latency for a demand
binary variables	
x_{ij}^k	1 if arc (i, j) is used by demand $k \in D$
z_{ift}^{kn}	1 if demand $k \in D$ uses copy n -th of VNF of type $f \in F$ placed on node $i \in N_c$ on a VM of type t
y_{ift}^n	1 if n -th copy of type of VM t is assigned to VNF of type $f \in F$ hosted by node $i \in N_c$
continuous non-negative variables	
ϕ_{ij}^k	flow for demand $k \in D$ on arc (i, j)
ψ_{if}^{kn}	flow for demand $k \in D$ entering in node i and using copy n of VNF of type $f \in F$
l_{if}^k	latency that demand $k \in D$ 'suffers' using VNF of type $f \in F$ hosted by node $i \in N_c$
c_i^k	order coefficient for node i in the path used by demand k

VNF compression/decompression linearization constraints:

$$\psi_{if}^{kn} \leq \sum_{j \in N: (j,i) \in A} \phi_{ji}^k + M_i (1 - \sum_{t \in T} z_{ift}^{kn}) \quad \forall i \in N_v, k \in D, f \in F, n \in C_f \quad (6)$$

$$\psi_{if}^{kn} \geq \sum_{j \in N: (j,i) \in A} \phi_{ji}^k - M_i (1 - \sum_{t \in T} z_{ift}^{kn}) \quad \forall i \in N_v, k \in D, f \in F, n \in C_f \quad (7)$$

$$\psi_{if}^{kn} \leq M_i \sum_{t \in T} z_{ift}^{kn} \quad \forall i \in N_v, \forall k \in D, \forall f \in F, n \in C_f \quad (8)$$

One compression/decompression VNF per node and demand:

$$\sum_{t \in T} \sum_{f \in F, n \in C_f: \mu_f \neq 1} z_{ift}^{kn} \leq 1 \quad \forall k \in D, \forall i \in N_v \quad (9)$$

Utilization rate constraints:

$$\sum_{k \in D} \phi_{ij}^k \leq U \gamma_{ij} \quad \forall (i, j) \in A \quad (10)$$

If there is no VNF, it cannot be used by any demand:

$$z_{ift}^{kn} \leq y_{ift}^n \quad \forall k \in D, \forall i \in N_v, \forall f \in F, \forall t \in T, n \in C_f \quad (11)$$

If traffic does not pass by a VNF, it cannot use it:

$$z_{ift}^{kn} \leq \sum_{j:(j,i) \in A} x_{ji}^k \quad \forall k \in D, \forall i \in N_v, \forall f \in F, \forall t \in T | m_{kf} = 1 \quad (12)$$

Each demand uses exactly one NFV of each type it asks for

$$\sum_{i \in N_c} \sum_{n \in C_f} \sum_{t \in T} z_{ift}^{kn} = 1 \quad \forall k \in D, \forall f \in F : m_{kf} = 1 \quad (13)$$

On each node at most a VM assigned for each VNF copy of a certain type:

$$\sum_{t \in T} y_{ift}^n \leq 1 \quad \forall f \in F, \forall i \in N_c, \forall n \in C_f \quad (14)$$

Node resource capacity (VNF utilization):

$$\sum_{f \in F} \sum_{n \in C_f} \sum_{t \in T} rr_{rt} y_{ift}^n \leq \Gamma_{ir} \quad \forall i \in N_v, \forall t \in T \quad (15)$$

Preventing the formation of isolated cycles:

$$c_j^k \geq c_i^k + x_{ij}^k - |A|(1 - x_{ij}^k) \quad \forall k \in D, \forall (i, j) \in A \quad (16)$$

Latency function linearization:

$$l_{if}^k \geq g_{ft}^j \left(\sum_{d \in D} \psi_{ift}^{dn} \right) - L_{max}(1 - z_{ift}^{kn}) \quad \forall i \in N_c, f \in F, n \in C_f \quad (17)$$

Maximum latency bound:

$$\sum_{(i,j) \in A} x_{ij}^k + \sum_{i \in N_c} \sum_{f \in F} l_{if}^k \leq L \quad \forall k \in D \quad (18)$$

Eq. (3) represents the flow balance for the access nodes. At destination node the quantity of flows is set equal to the demand multiplied for all factors of compression of all the demanded VNFs. Eq. (2) represents the flow balance for a given node that has the possibility of hosting VNFs.

We work under the assumption that given a node i , and a demand k , such demand uses at most a VNF f with a factor of compression/decompression $\mu_f \neq 1$. If a demand passes through a VNF with a factor of decompression μ_f , then the out-flow of the node is proportional to the in-flow:

$$\sum_{j \in N: (i,j) \in A} \phi_{ij}^k = \mu_f \sum_{j \in N: (j,i) \in A} \phi_{ji}^k$$

Using variable z that represents the assignment of demand to VNFs and subtracting the out-flow we get:

$$\sum_{j \in N: (i,j) \in A} \phi_{ij}^k - \sum_{j \in N: (j,i) \in A} \phi_{ji}^k = \sum_{j \in N: (j,i) \in A} \phi_{ji}^k \sum_{n \in C_f} \sum_{t \in T} (\mu_f - 1) z_{ift}^{kn} \quad (19)$$

The variable ψ_{if}^{kn} represents the flow of demand k that enters node i and passes through the copy n of the VNF of type f (non-linear representation):

$$\psi_{if}^{kn} = \left(\sum_{j \in N: (j,i) \in A} \phi_{ji}^k \right) \sum_{t \in T} \sum_{n \in C_f} z_{ift}^k$$

The constraints can be linearized using (6)-(8), with the parameter M_i equal to $\sum_{(j,i) \in A} \gamma_{ji}$, which represents the maximum quantity of flow entering node i .

Eq (4) and (5) allows to connect variables x and phi , in such a way that only and only if arc (i, j) is used by demand k , that is $x_{ij}^k = 1$, then variable phi can be different from zero. As the demand pass through VF that can compress or decompress the flow, then we can determine upper and lower bound for the demand that are: $b_k^{max} = b_k \prod_{f \in F: \mu_f \geq 1} \mu_f$ and $b_k^{min} = b_k \prod_{f \in F: \mu_f \leq 1} \mu_f$.

As mentioned before, we consider two objective functions:

- TE goal: minimize the maximum network link utilization:

$$\min U \quad (20)$$

- NFV goal: minimize number of cores (CPU) used by the instantiated VNFs:

$$\min \sum_{i \in N_v} \sum_{f \in F} \sum_{t \in T} \sum_{k \in D} \sum_{r \in CPU} rr_{rt} y_{ift}^n \quad (21)$$

The former objective allows taking into consideration the inherent fluctuations related to Internet traffic and therefore minimizing the risk of sudden bottleneck on network links. The latter assumes the fact that today the first modular cost in virtualization servers, especially in terms of energy consumption, is the CPU.

C. Multi-objective math-heuristic resolution

We face a multi-objective problem: minimizing the maximal link utilization and minimizing the total virtualization cost at the NFVI layer. These two objectives are in competition; in fact, to obtain a low utilization, a large number of VNFs must be allocated. We decided to prioritize the objectives. We minimize first the maximal link utilization, and then the NFV cost, which reflects the ISP-oriented vision to improve the user quality of experience (strictly related to link congestion, especially for real-time services).

In practice, in order to do our best to meet the second objective and allow the largest possible marginal gain, we perform a first optimization step to find the best solution accordingly to the first objective, and then, keeping the best value found in the first step as a parameter, we minimize the second objective. In fact, for a given optimal value of the first step, different possible configurations are available to the second step, and a large primary cost reduction can be achieved by this second step without losing with respect to the secondary objective. In order to reduce the computational time of the procedure, we performed an optimization procedure based on several steps. In the TE optimization, we subsequently optimize three models, starting from a simplified one, where no latency and compression/decompression features are present, to the

complete model. The solution of each step is given to the next optimization phase. In the NFV goal optimization, a bisection procedure is used on the number of allocated VMs to guarantee solution optimality, even when in a single step the solver is not able to guarantee it.

In order to allow a larger reduction of the total cost, at the price of a larger link utilization, an iterative approach can be used: increasing step by step the value of the first objective (TE) until the desired cost level of the second objective is found. Such an iterative procedure can have the advantage of providing to the second step of optimization a feasible solution (warm-start), which in many practical cases can reduce the computational time of the optimization.

IV. RESULTS

We implemented our VNF-PR algorithm using CPLEX and AMPL scripting. We limited the execution time to 300s for each TE optimization phase and to 800s for each the NFV optimization phase. In almost all tests we reached the optimum for both objectives. The average time needed for the overall procedure is 15 min. with a worst-case optimality gap of 10%.

We adopted the three-tier topology represented in Fig. 3 as a good reference for an Internet Service Provider network. Each access node is connected to two aggregation nodes, each aggregation node is connected to two core nodes, and core nodes are fully meshed. We consider all the nodes as NFVI nodes that can host VNFs. The demands are created uniformly in the interval $[a, b]$ so that edge demands cannot create any bottleneck at the access links. The aggregation links are dimensioned so that there is a risk of link saturation (i.e., link utilization higher than 100%) if the traffic distribution is not optimized. The core links are such that there is a very low bottleneck risk. Link latencies are set as follows, to cope for the different geographical scopes: 1 ms for access links, 3 ms for aggregation links, and 5 ms for core links (so that routing through core links is facilitated).

VNF processing latencies are set as in Fig. 2, for the fastpath and standard cases. We use two VM templates, one requiring 1 CPU and 16 GB of RAM, and one requiring 4 CPUs and 64 GB of RAM. We run tests setting for the end-to-end latency bound (L) with strict and loose values (15 and 20 ms, resp.). In order not to introduce unnecessary complexity to

capture the differences between the different cases, we limit to two VNF types per demand: a tunneling VNF (requiring decompression) and a firewall-like VNF. The NFVI layer is dimensioned so that there are enough computing resources to satisfy *individually* half of all the demands (i.e., excluding VNF sharing); NFVI access nodes are dimensioned so that they are composed of 5 CPUs, 80 GB RAM at each access node, twice and four times this quantity at aggregation and core nodes, respectively.

Each case is emulated with 10 random demand matrices. For the standard case, we analyze in the following the results shown in Fig. 4 (NFV cost distribution), Fig. 5 (link utilization distribution), Fig. 6 (end-to-end and VNF forwarding latency distribution).

We provide a detailed analysis in the following, with two points of views: what happens when we also consider the NFV cost in the objective function, and what happens when we make stronger the bound on the end-to-end latency. Then, we compare the standard case with the fastpath case.

A. TE vs. TE-NFV objectives sensibility

We analyze the difference between the results with the TE objective and the results with the combined TE-NFV objective.

- NFV cost (Fig. 4): the NFVI cost is significantly reduced with the TE-NFV objective, of roughly 70%. Indeed, a very high level of VNF sharing is enabled under the TE-NFV goal, for both latency bound situations.
- Link utilization (Fig. 5): even if the median utilization slightly increases passing from the TE goal to the TE-NFV goal, in particular for aggregation links (roughly 5% more), the overall link utilization distribution is not significantly affected by adding the NFV cost minimization in the optimization goal.
- Latency distribution (Fig. 6): the individual VNF processing latency contribution stays always below the ms, and a difference is barely visible using the TE-NFV approach. Thanks to the optimization, too high VNF sharing situations are avoided. This is a rather positive behavior considering that the Total latency (i.e., including propagation delays) is much higher, which could have hidden the importance of the VNF processing latency, which does not seem to be the case.

B. Sensibility to the latency bound

We analyze in the following the results highlighting the impact of the VNF chain latency bound L on the results.

- NFV cost (Fig. 4): the global NFV cost remains almost constant passing from the weak to the strong latency bound. Under the TE-NFV goal, we cannot detect any relevant change of behavior. Instead, under the TE goal, an interesting behavior is that, making weaker the latency bound ($L = 20ms$), more tunnelling VNFs are installed at the aggregation layer (less in the access and core layers), while more firewall VNFs are installed in the access and core layers. This behavior suggests that better utilization of high-capacity links (aggregation-core and

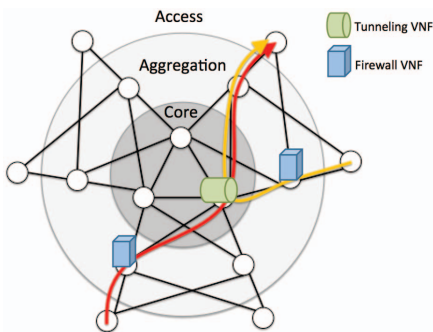


Fig. 3. Adopted network topology and VNF-PR solution example.

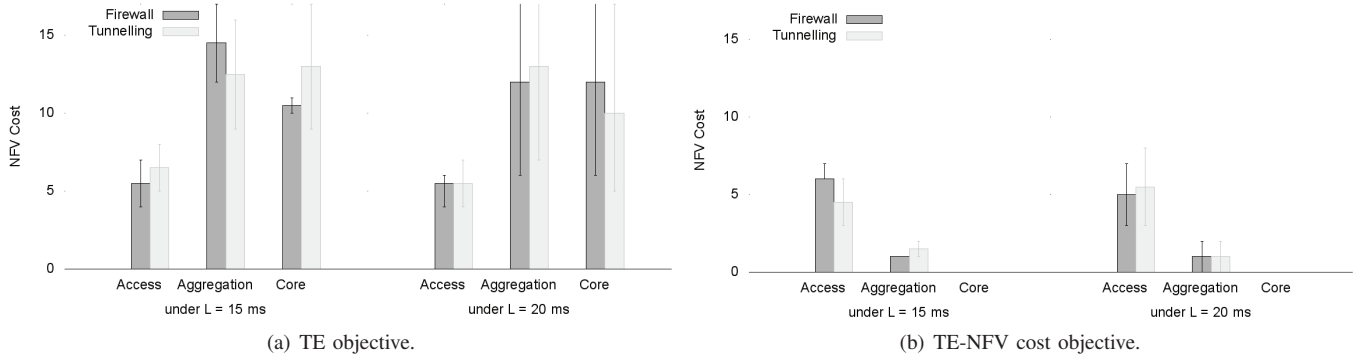


Fig. 4. VNF node distribution across NFVI layers (standard case).

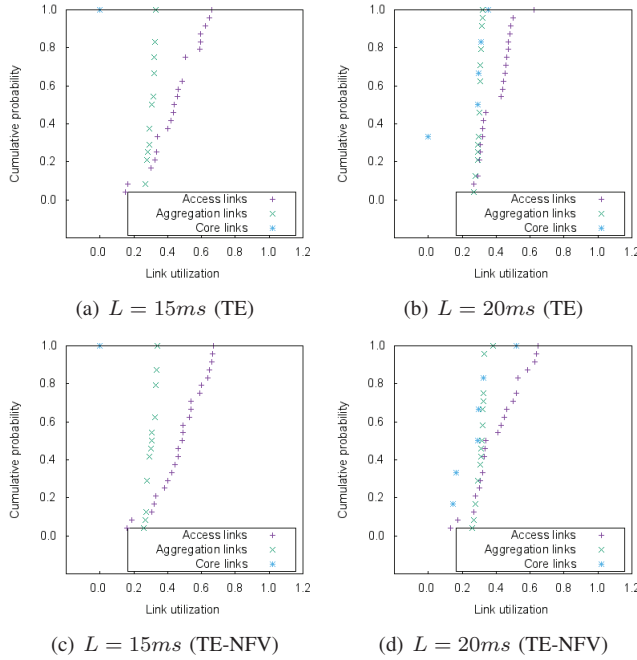


Fig. 5. Link utilization empirical CDFs (standard case).

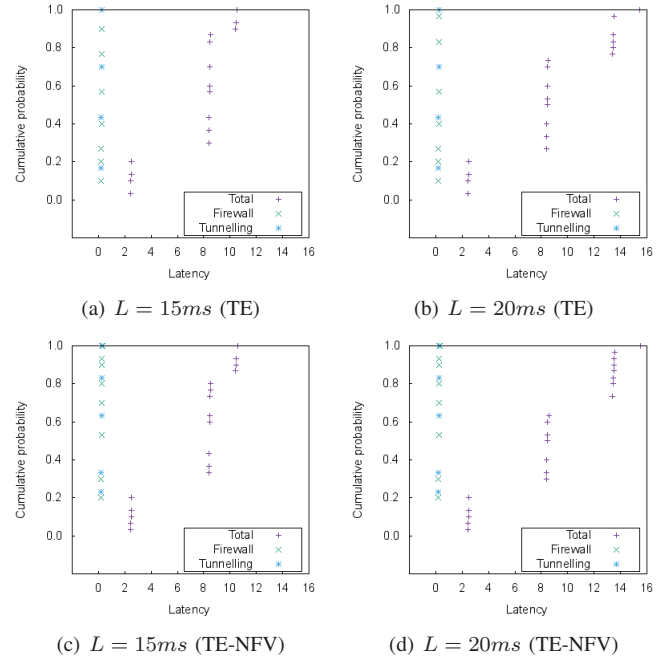


Fig. 6. Empirical CDFs of latency components (standard case).

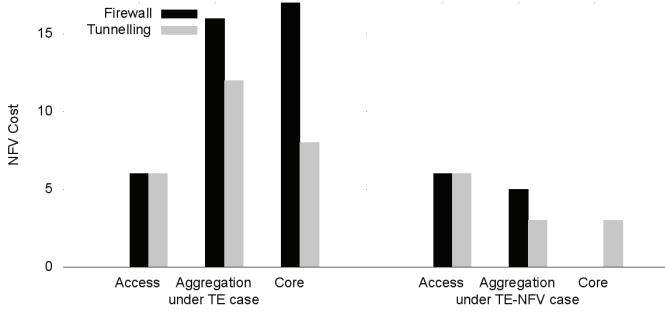
core-core links) can be granted by relaxing a bit the latency bound because the tunnelling VNFs cause a bit-rate increase, hence favoring the planning of firewall VNFs in access and core layers where more computing capacity is so made available.

- Link utilization (Fig. 5): in support of the above-mentioned analysis, we can remark that under the weak latency bound, core links get more utilized (with a median utilization increase of roughly 20%) while access links get less utilized (at maximum) when the NFV cost is not added to the optimization goal.
- Latency distribution (Fig. 6): the end-to-end latency bound has a direct visible impact only on the total latency, which confirms the importance of the bound only for VNF chaining decisions rather than for VNF placement decisions.

C. Standard vs. fastpath VNF switching

At last, we compare the standard case to the fastpath case. We consider only the weak latency bound ($L = 20$ ms) situation in order to better analyze the impact of the VNF forwarding mode (indeed, more stringent latency bounds have a relevant impact on the VNF chaining only). Fig. 7, Fig. 8 and Fig. 9 show respectively the NFV cost, link utilization, latency distributions, to be compared with the corresponding previous ones. We can observe that:

- NFV cost (Fig. 7 vs. Fig. 4): fastpath VNF forwarding is more expensive in terms of NFV cost than the standard one, under both optimization goals. It is roughly 20% more expensive. This is a direct result of the maximum traffic bound that is set under the fastpath case and that is not set for the standard case. The value itself of the bound on the maximum traffic that can be forwarded is not that relevant (even if it can be easily tuned in practice),

Fig. 7. VNF node distribution across NFVI layers (fastpath, $L = 20ms$).

rather its existence logically always makes fastpath more expensive than legacy standard NFV deployment with piece-wise forwarding latency behavior. This is the other side of the cost-performance trade-off, fastpath offering the better forwarding performance. On the other hand, the VNF node distribution shows that also with fastpath the core layer is much less used under the TE-NFV goal.

- Link utilization (Fig. 8 vs. Fig. 5): a similar behavior than the one detected for the standard case can be remarked.
- Latency distribution (Fig. 9 vs. Fig. 6): surprisingly, it appears that VNFs are better shared (higher VNF forwarding latency) under the fastpath forwarding mode, with a latency close to the maximum allowed by the second template. This is likely a side, yet desirable, effect of the already mentioned bound on the maximum traffic to be forwarded by the VNF, and it could be considered as a desirable NFVI network management property.

V. CONCLUSION

This paper proposes a VNF chaining and placement model, including an algorithm formulated as a mixed integer linear program, which goes beyond recent work at the state of the art. Our model takes into consideration specific NFV forwarding modes (standard and fastpath modes) as well as flow bitrate variations that make the allocation of edge demands over VNF chains unique yet complex. In order to master the time complexity while considering two different optimization goals – traffic engineering (TE) goal alone and TE goal combined with NFV infrastructure cost minimization goal (TE-NFV) – we adopt a math-heuristic resolution method.

We run extensive tests to evaluate our algorithm on a three-tier topology representing an ISP topology. The results show that the combined TE-NFV objective significantly reduces the number of VNFs in the network compared to the TE objective with almost no impact on the link utilization and on the latency. The bound on the end-to-end latency mainly affects the utilization of the links, core links being more utilized for weak bounds. The relaxation of this bound has another interesting effect for the TE objective: it pushes tunnelling VNFs, that increase the bitrate, at the aggregation level and firewall-like VNFs at the access and core levels.

Another important insight of our analysis is that, using a fastpath forwarding mode (i.e., a VNF setting that allows

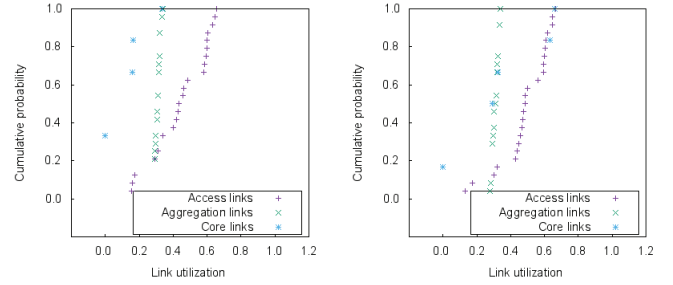
(a) $L = 20ms$ (TE)(b) $L = 20ms$ (TE-NFV)

Fig. 8. Link utilization empirical CDFs (fastpath case).

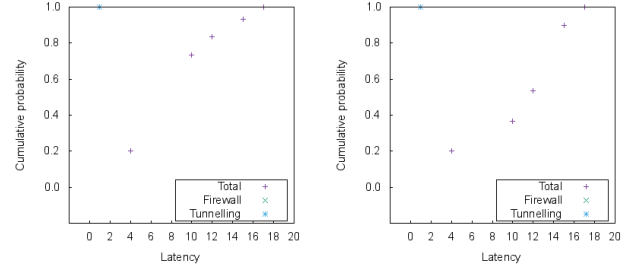
(a) $L = 20ms$ (TE)(b) $L = 20ms$ (TE-NFV)

Fig. 9. Empirical CDFs of latency components (fastpath case).

a constant-latency forwarding time up to a maximum load) allows higher virtual network function resource sharing than using a standard forwarding mode (leading to a piece-wise forwarding latency behavior), with a 20% higher infrastructure cost.

ACKNOWLEDGMENT

This work was partially supported by the ANR Reflexion project (<http://anr-reflexion.telecom-paristech.fr>, contract nb: ANR-14-CE28-0019), and the European Institute of Technology (EIT) Digital Future Networking Service Action Line. We thank Nicolas Roubertier for helping with the simulations.

REFERENCES

- [1] M. Chiosi and et al. Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action. In *SDN and OpenFlow World Congress*, 2012.
- [2] ETSI. Network Functions Virtualization - Introductory White Paper. October 2012.
- [3] Intel. Impact of the Intel Data Plane Development Kit (Intel DPDK) on packet throughput in virtualized network elements. White Paper, 2009.
- [4] R. Guerzoni and et al. A novel approach to virtual networks embedding for SDN management and orchestration. In *IEEE/IFIP NOMS 2014*.
- [5] S. Mehraghdam, M. Keller, and K. Holger. Specifying and placing chains of virtual network functions. In *IEEE CLOUDNET 2014*.
- [6] H. Moens and F. de Turck. VNF-P: A model for efficient placement of virtualized network functions. In *CNSM 2014*.
- [7] R. Mijumbi and et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions. In *IEEE NETSOFT 2015*.
- [8] M. Xia and et al. Network function placement for NFV chaining in packet/optical datacenters. In *ECOC 2014*.
- [9] M. Bouet, J. Leguay, and V. Conan. Cost-based placement of vDPI functions in NFV infrastructures. In *IEEE NETSOFT 2015*.
- [10] Intel. Network Function Virtualization Packet Processing Performance of Virtualized Platforms with Linux* and Intel Architecture. Technical Report, Oct. 2013.