Optimal Service Function Chain Composition in Network Functions Virtualization

Andrés F. Ocampo, Juliver Gil-Herrera, Pedro H. Isolani, Miguel C. Neves, Juan F. Botero, Steven Latré, Lisandro Zambenedetti, Marinho P. Barcellos, Luciano P. Gaspary

Optimal Service Function Chain Composition in Network Functions Virtualization

Andrés F. Ocampo[1], Juliver Gil-Herrera[1], Pedro H. Isolani[2], Miguel C. Neves[3], Juan F. Botero[1], Steven Latré[2], Lisandro Zambenedetti[3], Marinho P. Barcellos[3], and Luciano P. Gaspary[3]

[1] University of Antioquia, Cl. 67 # 53 - 108 - Medellín, Colombia,
{andres.ocampop, juliver.gil, juanf.botero}@udea.edu.co
[2] University of Antwerp - imec, Middelheimlaan 1, 2020 Antwerp, Belgium,
{pedro.isolani, steven.latre}@uantwerpen.be
[3] Federal University of Rio Grande do Sul, Paulo Gama, 110 - Porto Alegre, Brasil,
{mcneves, granville, marinho, paschoal}@inf.ufrgs.br

Abstract. Network Functions Virtualization (NFV) is an emerging initiative where virtualization is used to consolidate Network Functions (NFs) onto high volume servers (HVS), switches, and storage. In addition, NFV provides flexibility as Virtual Network Functions (VNFs) can be moved to different locations in the network. One of the major challenges of NFV is the allocation of demanded network services in the network infrastructures, commonly referred to as the Network Functions Virtualization - Resource Allocation (NFV-RA) problem. NFV-RA is divided into three stages: (i) Service Function Chain (SFC) composition, (ii) SFC embedding and (iii) SFC scheduling. Up to now, existing NFV-RA approaches have mostly tackled the SFC embedding stage taking the SFC composition as an assumption. Few approaches have faced the composition of the SFCs using heuristic approaches that do not guarantee optimal solutions. In this paper, we solve the first stage of the problem by characterizing the service requests in terms of NFs and optimally building the SFC using an Integer Linear Programming (ILP) approach.

Keywords: Network Function Virtualization, Virtual Network Functions, Service Function Chain, VNFs Chain Composition

1 Introduction

Network Functions Virtualization is an emerging network management framework for service deployment, which allows Network Functions to be allocated onto general purpose servers [2]. It enables to dynamically compose chains of Virtual Network Functions and embed them anywhere in the network according to a predefined objective. For instance, network functions such as firewalls, load balancers, and deep packet inspection systems can be placed at the most appropriate location in the network to support users demand, Quality of Service (QoS) requirements, or management needs. NFV has grabbed the attention

from industry because it has the potential to reduce both CAPEX and OPEX, by the dynamic deployment of VNFs to commodity hardware, avoiding vertically integrated solutions. In academia, NFV has been a hot topic because there are interesting technical challenges to be overcome [1–3], such as the NFV allocation problem [4, 5].

To better understand our proposed model in later sections, we introduce the most important terms used throughout this paper.

Network Service (NS): It is an offering provided by an operator that is delivered using one or more network functions. Network service is a complete, end-to-end functionality provided by the network operator, such as network protection. A network service may comprise one or more VNFs, for example, a firewall, a deep packet inspector (DPI), and a data monitor, as in the case of a network protection system.

Virtual Network Function (VNF): It is a function responsible for a specific treatment of data flows. A VNF can act at various network layers of the protocol stack. As a logical component, a VNF can be realized as a virtual element or be embedded in a physical network appliance. One or more VNFs can be embedded in the same physical element.

Service Function Chain (SFC): It is an ordered or partially ordered set of VNFs. The implied order may not be a straight line, since the architecture allows SFCs that send traffic to more than one branch, and also allows cases where there is flexibility in the order in which VNFs need to be applied. SFCs may be unidirectional or bidirectional, depending on the state requirements of the network functions. Many common functions such as DPI and firewalls often require bidirectional chaining in order to ensure that the flow state is consistent. An SFC, in ETSI's terminology, is called VNF Forwarding Graph (VNF-FG)[4].

Efficient network services require the optimal allocation of resources in NFV (NFV-RA), a challenging problem [5]. A chain of VNFs must be intelligently composed and allocated to the infrastructure to provide end-to-end QoS guarantees for the applications. However, given the VNFs dependencies, the allocation is extremely challenging.

The above mentioned NFV-RA problem can be sub-divided into three sub-problems: (i) SFC composition, (ii) SFC embedding and (iii) SFC scheduling. Due to the fact that several chains can fulfill the same NS, the order of VNFs is often flexible; that is, some VNFs have to be placed in a specific order (e.g., the network flow first has to be decrypted before it can be further processed), while others are flexible in that regard (i.e. they dont depend from one another). Therefore, the composition of the best possible chain (SFC composition) for each NS is very important for the operator. However, SFC composition has been so far overlooked by the scientific community, typically taken as an assumption. Besides, to the best of our knowledge, previous solutions are heuristic in nature and, therefore, do not guarantee optimal solutions.

In this paper, our contributions are twofold: 1) we propose a way to formally describe network services as a set of VNFs considering the dependences among

[4] http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf

them and how they can be concatenated in SFCs and 2) we propose an ILP-based approach to optimally solve this sub-problem by characterizing the service requests in terms of virtual network functions and solving the SFC composition problem.

The remainder of this paper is organized as follows. In Section 2, we describe the main approaches tackling the NFV-RA problem. In Section 3, we define the SFC composition problem. Section 4 specifies our ILP formulation in detail. Section 5 presents the performance evaluation of our proposed approach. Finally, in Section 6, we conclude the paper with final remarks and perspectives for future work.

2 Related Work

The majority of current NFV-RA approaches starts from the assumption that the chain of VNFs has been already composed, *i.e.*, the important stage of SFC composition is taken for granted. Few approaches have been proposed to solve the SFC composition stage so far. Mehraghdam *et al.* [7] formulate a context-free language for formalizing chaining requests. They propose a greedy heuristic that tries to minimize the total data rate of the resulting chain by composing first the VNF that reduces the data rate of the flows in each step. Recently, Beck and Botero [1] proposed a scalable recursive heuristic that, at each step, composes a VNF in the service chain and, at the same time, embeds it in the substrate network (SN) trying to rapidly find a feasible solution.

Most of the existing NFV-RA approaches deal just with the embedding stage as they consider the VNF-FG as a given input of the problem [5]. For instance, Bari *et al.* [4] propose exact Integer Linear Programming (ILP) and heuristic based approaches trying to minimize the OPEX caused by the SFC embedding. Also, Elias *et al.* [3] formulate the SFC embedding as a non-linear integer optimization model where the objective function is to minimize the network congestion.

The aforementioned review shows that, up to now, little research has been performed in the composition stage of the SFC problem. Current solutions are heuristic-based and no optimal solution for the problem has been proposed so far. An optimal solution results in the best possible composition of the service chain with regard to a predefined objective. In this paper, we propose an optimal approach to solve the problem based on Integer Linear Programming.

3 The SFC Composition Problem

When allocating resources for a given NS, service providers receive a chain of VNFs and apply an embedding strategy for placing and linking these functions on the physical substrate. Despite being automatic, this process is rigid for clients and service providers, respectively. While clients must deal with complex function dependencies when specifying services, providers are not able to structure the chains in order to find the best fit for their infrastructure. The result is the

allocation of suboptimal chains, which may require many more VNF instances or network bandwidth than necessary, leading to high costs and expenditures.

Although some dependencies and connections among VNFs in a service must be considered, the order of the VNFs (*i.e.*, the structure of the chain) is often flexible. For example, normally there is no explicit dependency between a leakage prevention system and a traffic shaper or between a proxy server and a WAN optimizer. As a consequence, it is possible for several different chains to fulfill the same service. We call the problem of finding the most appropriate VNF chain, given a network service specification and a set of resource constraints, the *SFC composition problem*.

Figure 1(a) shows our proposal on how a network service specification (*i.e.*, a Virtual Network Functions Request - VNFR) looks like. Instead of providing the VNF chain structure as a whole, clients have to inform only the necessary information for allowing network service providers to derive the best chain according to some predefined goal (*e.g.*, to minimize the number of NF instances or the bandwidth demand).

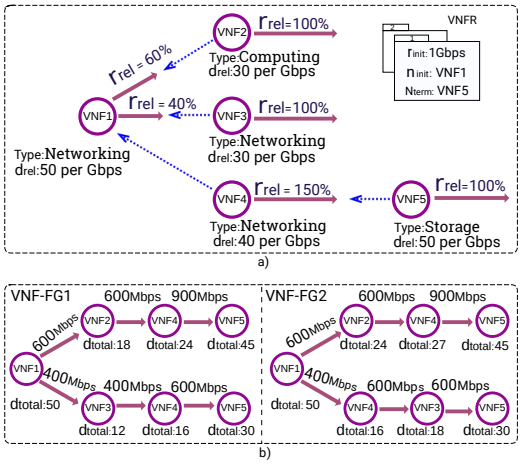


Fig. 1: VNF chain composition

Essentially, a VNFR has five elements: *(i)* the initial data rate of the network flow (r_{init}), *(ii)* the set of VNFs that compose the service, each one with their respective processing requirement (d_{rel}), *(iii)* the VNFs where the flow initiates (n_{init}) and terminates (n_{term}), *(iv)* a number of outgoing links (solid purple arrows) at each VNF, and *(v)* mandatory dependencies (dotted blue arrows).

Outgoing links can be used to represent scenarios where traffic is split (*e.g.*, bifurcations). If a VNF has more than one outgoing link, then it splits the traffic flow into the same number of sub-flows. For example, consider a load balancer or a DPI server separating TCP and non-TCP traffic. Each sub-flow has a relative

traffic rate (r_{rel}), which can be higher than 100% if the function replicates or encodes traffic.

Dependencies, in turn, may be of two different types: between a VNF and an outgoing link, or between two distinct VNFs. Outgoing link dependencies represent VNFs that should selectively be placed on one of the sub-flows (*e.g.*, a firewall that succeeds an anomaly-based IDS only for suspicious traffic). Dependencies between VNFs, on the other hand, indicate that the dependent function must be present in each and every sub-flow in the chain (*e.g.*, the cache servers succeeding a load balancer).

In Figure 1(b), we represent two possible chains for the service described in Figure 1(a). Notice that VNFs 2 and 3 selectively appear in the sub-flows of VNF 1, while VNF 4 is present in both subpaths. Moreover, bandwidth and processing demands are determined according to the relative traffic of each outgoing link. Although structurally similar, the left chain (*i.e.*, VNF-FG 1) requires less network and computing resources, which at the end results in lower costs for both clients and providers. As an outcome of the SFC composition problem, VNF-FG 1 would be sent for embedding in the provider infrastructure.

Table 1 details the notation used in the model proposed in the following section. The first part of the table introduces the parameters of a Network Service Request. The second part explains the sets of nodes of an augmented graph used to build the ILP model. Finally, the table shows the ILP variables.

Before jumping into the next section, it is worth mentioning that the outcome of the chain composition stage is one complete service chain (VNF-FG) with regard to one predefined objective and that the amount of required capacities depends on the amount of data handled by that VNF instance.

4 SFC Composition Problem Formulation

4.1 Augmented Graph

To solve the SFC composition problem, we propose an ILP model that is built based on an augmented graph created from the VNFR as follows:

- For the first VNF (n_{init}) of the VNFR, one node is placed in the augmented directed graph $G^{ext} = (V^{ext}, L^{ext})$;
- For each of the remaining VNFs, we create as many nodes as the maximum number of instances that a VNF may have. For example, in Figure 2, the maximum number of instances is 2, as the VNF 1 splits the traffic in two sub-flows. To ease the notation, the node $i^m \in V^{ext}$ denotes the m-th instance of the node $i \in V$;
- Then, we place directed links between each pair of nodes of the graph except for:
  - Instances of the same VNFs: $(i^m, j^n) \in L^{ext}, \forall i^m \in V^{ext}, j^n \in V^{ext} \iff i \neq j$;
  - Links directed to n_{init}: $(i^m, j^n) \in L^{ext}, \forall i^m \in V^{ext}, j^n \in V^{ext} \iff j \neq n_{init}$.

Table 1: SFC Composition Inputs and Variables

Virtual Network Functions Request (VNFR)	$VNFR(V, L)$	Is the service request formed by V VNFs and L VNF links	Sets
	V	Set of VNFs	
	L	Set of VNF links; this is the set of all links coming out the different VNFs belonging to the VNFR	
	$L^i_{out} \subset L$	Determines the (outgoing) VNF links of the VNF $i \in V$; a VNF with multiple links splits the network flow into several sub-flows	
	$N_{term} \subset V$	VNFs where the service terminates	
	$r_{init} : \mathbf{Z}$	Initial data rate of the VNFR	Parameters
	$r_{rel}(i) : \mathbf{V} \rightarrow \mathbf{Z}$	Total traffic (in percentage) departing from node $i \in V$	
	$n_{init} \in V$	Initial VNF of the service	
	$d_{rel}(i) : V \rightarrow \mathbf{R}$	Relative processing capacity demands of the VNF $i \in V$	
	$r_{rel}(i, b) : L \rightarrow \mathbf{Z}$	Relative link traffic rate of link $(i, b) \in L$, here $i \in V$ identifies the link source VNF and b is the link number of i	
	$req(i) : V \rightarrow L^*$	Dependencies of the VNF $i \in V$; defined as the incoming edges of a VNF and refer to outgoing links of other VNFs. This allows the specification of VNFs that get selectively deployed on specific sub-flows. An assignment of a VNF instance is only valid if traffic has first been routed through instances of all the required VNFs.	
	$MI(i)$	Minimum number of instances of the VNF $i \in V$ in the VNF-FG	
Augmented Graph	$G^{ext} = (V^{ext}, L^{ext})$	This augmented graph is created from the VNFR. The final service chain (VNF-FG) will be a subgraph of G^a	Sets
	V^{ext}	Set of nodes of the augmented graph, each node in V may have one or more instances in V^{ext}	
	L^{ext}	Set of links of the augmented graph	
	P	Set of paths from the node in V^{ext} that correspond to the instance of $n_{init} \in V$ to the set of instances of the terminating nodes N_{term} in the augmented graph $i^m \in V^{ext} : i \in N_{term}$	
	$Pos^p_{i^m}$	Position of the augmented node $i^m \in V^{ext}$ in path $p \in P$, if i^m is not part of the path, then 0	Parameters
	$\delta^p_{i^m, j^n}$	Binary parameter that indicates if augmented link $(i^m, j^n) \in L^{ext}$ is part of path p	
	$y^{i,b}_{i^m, j^n}$	Binary variable that says whether the link $(i, b) \in L$ is mapped in the link $(i^m, j^n) \in L^{ext}$ of the augmented network	Variables
	y_{i^m, j^n}	Binary variable that says whether the link $(i^m, j^n) \in L^{ext}$ of the augmented network is chosen as a part of the resulting VNF-FG	
	x^m_i	Binary variable that says whether the instance $i^m \in V^{ext}$ of the augmented network is part of the resulting VNF-FG	
	$LD^{i,b}_{i^m, j^n}(BW)$	Bandwidth required by link $(i, b) \in L$ assigned to $(i^m, j^n) \in L^{ext}$ in the augmented network	
	$LD_{i^m, j^n}(BW)$	Total bandwidth required by $(i^m, j^n) \in L^{ext}$ in the augmented network	
	$TD_{i^m}(BW)$	Total bandwidth arriving to node (i^m) in the augmented network	
	$z^{i,b}_{i^m, j^n}$	Auxiliary variable to perform the following product between variables $z^{i,b}_{i^m, j^n} = TD_{i^m}(BW) \cdot y^{i,b}_{i^m, j^n}$	
	$y^{i,b,p}_{i^m, j^n}$	Binary variable that says whether the link $(i, b) \in \mathbf{L} : i \in V^k$ is mapped in the link (i^m, j^n) belonging to the path p of the augmented network	
	u^p	Binary variable that says whether the path $p \in P$ is used in the augmented network	

Figure 2, where n_{init} =VNF1, shows how an augmented graph (see Figure 2b[5]) is created from a VNFR (see Figure 2a).

Our ILP model is based on the fact that any possible chaining is a subgraph of the augmented graph, so the ILP variables (cf. Table 1) mainly denote which nodes and links of the augmented graph are considered to be parts of the chain, and the demands they will have due to the chosen chaining. Figure 3a shows two possible chains that can be created out of VNFR in Figure 2a, and how they are present in the augmented graph (see Figure 3b).

[5] For the sake of clarity, directed links are drawn with arrows, so a link with arrows in its extremes a and b represents a pair of directed links; one from a to b, and the other from b to a

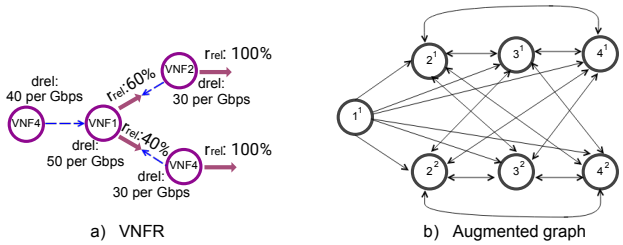


Fig. 2: VNFR and Augmented Graph

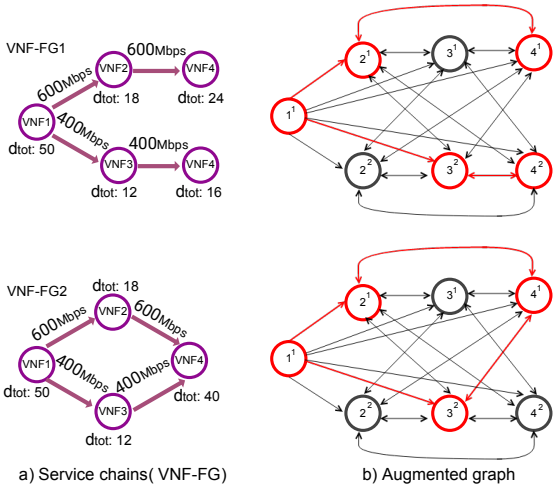


Fig. 3: VNFR and Augmented Graph

4.2 ILP Formulation

The following is the ILP formulation of the chain composition problem. That is to say, the ILP that models how to insert a VNFR in the augmented graph with respect to a predefined objective (*e.g.* minimize the number of VNF instances of the resulting VNF-FG).

Constraints:

Link Mapping Constraints:

$$y^{i,b}_{i^m,j^n} \leq \sum_{p \in P} y^{i,b,p}_{i^m,j^n} : (i^m, j^n) \in L_{ext}, (i,b) \in L \tag{1}$$

$$H \cdot y^{i,b}_{i^m,j^n} \geq \sum_{p \in P} y^{i,b,p}_{i^m,j^n} : (i^m, j^n) \in L_{ext}, (i,b) \in L \tag{2}$$

$$y^{i,b,p}_{i^m,j^n} \leq \delta^{p}_{i^m,j^n} : (i^m, j^n) \in L_{ext}, (i,b) \in L, p \in P \tag{3}$$

Equations 1, 2, and 3 indicate the relationship between variables $y^{i,b}_{i^m,j^n}$ and $y^{i,b,p}_{i^m,j^n}$ (here H is a big number that enforces the binary constraint). They indicate that the augmented link $(i^m, j^n) \in L^{ext}$ can map the outgoing link $(i, b) \in L$ when it is mapped using a predefined path $p \in P$. Remember that the same outgoing link $(i, b) \in L$ may be mapped in several links $(i^m, j^n) \in L^{ext}$. For example, in Figure 1a, in the VNFR, the outgoing link of VNF 4 is mapped in two different links between VNF 4 and VNF 5 in the VNF-FG 1 (see Figure 1b).

Equations to ensure that if a path p is mapped to the augmented graph, all its links have to be assigned.

$$\sum_{(i^m,j^n)\in L_{ext}} \sum_{(i,b)\in L} y^{i,b,p}_{i^m,j^n} \leq H \cdot u^p : p \in P \tag{4}$$

$$\sum_{(i,b)\in L} y^{i,b,p}_{i^m,j^n} \geq \delta^p_{i^m,j^n} \cdot u^p : (i^m, j^n) \in L_{ext}, p \in P \tag{5}$$

Equation 4 ensures that if a path is not mapped in the augmented graph, then variable $y^{i,b,p}_{i^m,j^n}$ should be zero for all links belonging to that path. Equation 5 ensures that if path is mapped in the augmented graph, then the variable $y^{i,b,p}_{i^m,j^n}$ should be one for all links belonging to that path.

Equation 6 ensures that an outgoing link $(i, b) \in L$ of the VNFR should be mapped in an augmented link of the augmented graph just for one path:

$$\sum_{p\in P} y^{i,b,p}_{i^m,j^n} \leq 1 : (i^m, j^n) \in L_{ext}, (i, b) \in L \tag{6}$$

Establishment of y_{i^m,j^n}:

$$y_{i^m,j^n} = \sum_{(i,b)\in L} y^{i,b}_{i^m,j^n} \leq 1 : (i^m, j^n) \in L_{ext} \tag{7}$$

$$y_{i^m,j^n} \leq x^m_i : (i^m, j^n) \in L_{ext} \tag{8}$$

$$y_{i^m,j^n} \leq x^n_j : (i^m, j^n) \in L_{ext} \tag{9}$$

Constraints 7, 8, and 9 establish the belonging of a link of the augmented graph $(i^m, j^n) \in L^{ext}$ to the resulting service chain (VNF-FG).

Node Mapping Constraints:

Establishment of x^m_i:

$$x^m_i = \sum_{(i^m,j^n)\in L_{ext}} y^{i,b}_{i^m,j^n} : i \in V^k, 1 \leq m \leq M^i, (i, b) \in L^i_{out} \tag{10}$$

Lower bound in the possible number of instances for i:

$$\sum_{m=1}^{M_i} x^m_i \geq MI(i) : i \in V^k \tag{11}$$

Constraints 10 and 11 establish the belonging of a node $i^m \in V^{ext}$ in the resulting service chain (VNF-FG).

Dependencies fulfillment constraints:

$$\sum_{(i^m,j,n)\in L_{ext}} y^{i,b}_{i^m,j^n} \geq 1 : l \in V^k, l \neq n_{init}, (i,b) \in req(l) \tag{12}$$

Constraint 12 ensures that for each node $l \in V$, all the dependencies are mapped.

Position constraints:

$$y^{i,b,p}_{i^m,j^n} \cdot Pos^p_{i^m} \leq Pos^p_{l^r} : l \in V^k, l \neq n_{init}, (i,b) \in req(l), \\ l^r \in V_{ext}, (i^m,j^n) \in L_{ext}, p \in P, \delta^p_{i^m,j^n} \neq 0 \tag{13}$$

$$\sum_{l^r \in V_{ext}} \sum_{p \in P : l^r \in P} \sum_{(i^m,j^n)\in L_{ext}} y^{i,b,p}_{i^m,j^n} \geq 1 : l \in V^k, l \neq n_{init}, (i,b) \in req(l) \tag{14}$$

Constraint 13 ensures the precedence of the dependencies. If one VNF instance $l \in V$ is mapped in the augmented graph, then the set of its dependent links should be mapped in a prior position in the path going from n_{init} to l. Constraint 14 ensures that the path being used to map the VNF's dependencies also contains an instance of the VNF.

Incoming links for each x^m_i:

$$\sum_{(j^n,i^m)\in L_{ext}} y_{j^n,i^m} \geq x^m_i : i \in V^k, 1 \leq m \leq M^i, i \neq n_{init} \tag{15}$$

$$\sum_{(j^n,i^m)\in L_{ext}} y_{j^n,i^m} \leq H \cdot x^m_i : i \in V^k, 1 \leq m \leq M^i \tag{16}$$

Equations 15 and 16 state that if a link of the augmented graph is part of the resulting service chain (VNF-FG) then the end nodes of this link should be part of the chain too.

The following Equations to establish $z^{i,b}_{i^m,j^n}$ linearize the following product $z^{i,b}_{i^m,j^n} = TD_{i^m}(BW) \cdot y^{i,b}_{i^m,j^n}$.

$$z^{i,b}_{i^m,j^n} \leq y^{i,b}_{i^m,j^n} \cdot H : (i^m,j^n) \in L_{ext}, (i,b) \in L \tag{17}$$

$$z^{i,b}_{i^m,j^n} \leq TD_{i^m}(BW) : (i^m,j^n) \in L_{ext}, (i,b) \in L \tag{18}$$

$$z^{i,b}_{i^m,j^n} \geq TD_{i^m}(BW) - (1 - y^{i,b}_{i^m,j^n}) \cdot H : (i^m,j^n) \in L_{ext}, (i,b) \in L \tag{19}$$

Constraints to set demands:

$$LD^{i,b}_{i^m,j^n}(BW) = r_{rel}(i,b) z^{i,b}_{i^m,j^n} : (i^m, j^n) \in L_{ext}, (i,b) \in L \tag{20}$$

$$LD_{i^m,j^n}(BW) = \sum_{(i,b)\in L} LD^{i,b}_{i^m,j^n}(BW) : (i^m, j^n) \in L_{ext} \tag{21}$$

$$TD_{j^n}(BW) = \sum_{(i^m,j^n)\in L_{ext}} LD_{i^m,j^n}(BW) : j^n \neq n_{init} \in V_{ext} \tag{22}$$

$$TD_{n^1_{init}}(BW) = r_{init} \cdot x_{n^1_{init}} \tag{23}$$

$$\sum_{(i^m,j^n)\in L_{ext}} LD_{i^m,j^n}(BW) = TD_{i^m}(BW) \cdot r_{rel}(i) : i^m \in V_{ext}, i \notin N_{term} \tag{24}$$

These set of equations simply set the bandwidth demand of each link in the augmented graph and also the complete load received by each node in the augmented graph. Here, H is just a big number to force binary variables to take 0 or 1 values.

5 Performance Evaluation

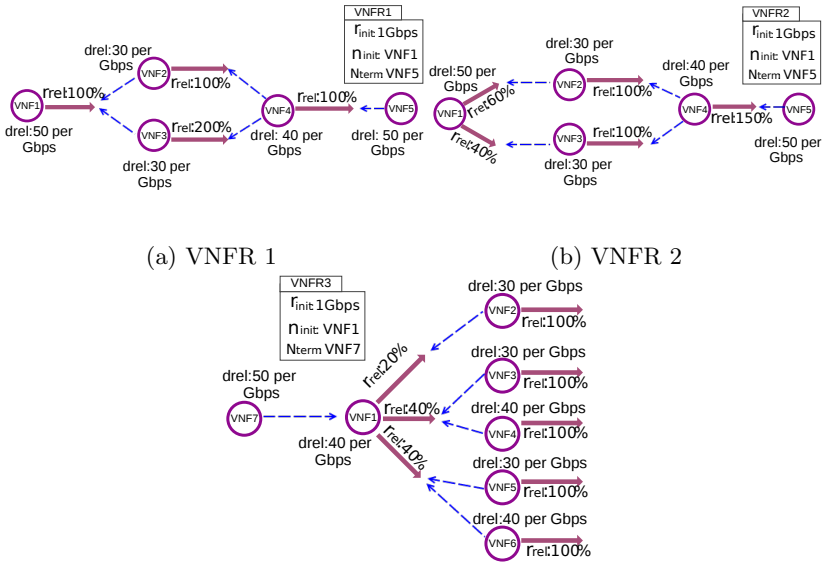
In this section, a performance evaluation of the ILP is presented. Our evaluation focuses primarily on minimizing the total bandwidth demanded by the constructed service chain (VNF-FG). Three scenarios are configured in order to analyze our ILP model following typical cases for service chaining [8].

5.1 Simulation Scenario

The ILP model for SFC composition presented in the previous section was implemented in the Gurobi solver [6] which provides an exact solution. To the best of our knowledge, just one work [7] has dealt separately with the composition phase of NFV-RA. This work heuristically tries to allocate those VNFs with flexible order following an ascending order according to their ratio of outgoing to incoming data rate. Here, we compare our ILP-based exact model with this heuristic proposal.

Simulations are performed for three VNFRs based on typical use cases of networks chains [8]. Figure 4 illustrates the simulation settings for each scenario. The first request VNFR 1 is given for service with NAT64 functions where the traffic is processed by a subchain (composed of VNF 1, VNF 2 and VNF 3), then the NAT function (VNF 4) for IP capabilities (e.g., mapping from IPv6 to IPv4), and then processed by another subchain (VNF 5). The first subchain could have VNFs with optional order, so a good planning of such functions in the final VNF-FG would impact the entire network performance.

The second scenario (VNFR 2) follows the structure of a service chain used to split service paths where service providers enable content awareness. VNF 1 splits the traffic into two sub-flows through two outgoing links. On the one



(a) VNFR 1 (b) VNFR 2

(c) VNFR 3

Fig. 4: Virtual Network Function Requests

hand VNF 2 is disposed at the sub-flow of VNF 1 to process 60% of its outgoing traffic; on the other hand, VNF 3 is located at the second sub-flow of VNF 1 to process 40% of its outgoing traffic. VNF 4 has to be processed by both sub-flows and, subsequently, the final function of the VNFR is VNF 5.

Finally, VFNR 3 is given for scenarios in the Gi Interface for mobile network environments. We define an scenario with seven VNFs disposed as follows: VNF 1 is the initial function to be performed, this function divides the incoming traffic into three sub-flows through three links with relative bandwidth demands of 20%, 40% and 40% respectively. VNF 2 depends on the first sub-flow while VNFs 3 and 4 depend on the second sub-flow and VNFs 5 and VNF 6 depend on the last sub-flow. Finally, Each sub-flow must to be processed by the terminal function VNF 7.

5.2 Results

The solution of our model is given in terms of a VNF-FG to be embedded on the physical network. The main objective is to find a VNF-FG with the minimal bandwidth demand on its links. Therefore, the objective function of our ILP is to minimize:

$$\sum_{(i^m,j^n)\in L_{ext}} LD_{i^m,j^n}(BW) \tag{25}$$

In order to validate the effectiveness of our solution, we compare it with the heuristic mentioned before, in terms of the total bandwidth demanded by VNF-FG, that is, the sum of all bandwidth demands on each link of the VNF-FG.

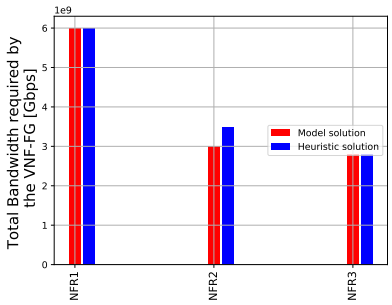


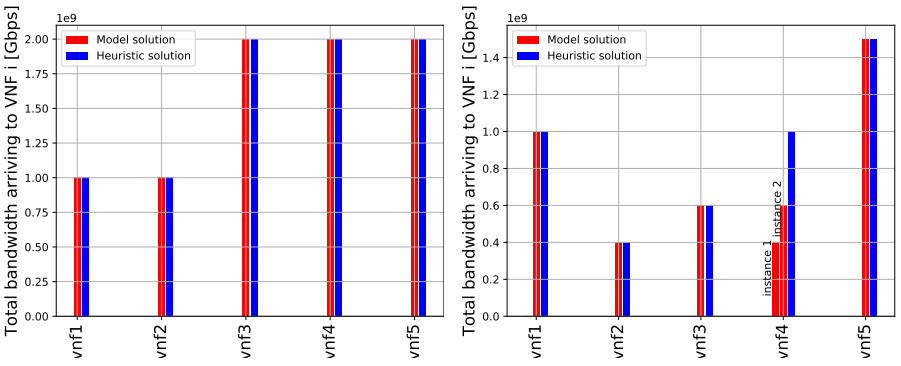
Fig. 5: Total bandwidth demanded by VNF-FG

Figure 5 shows the comparison between the ILP model and the heuristic in the aforementioned scenarios. On the one hand, for scenarios where the traffic is split out into several links (e.g. VNFR 2), our solution yields better results than the VNF-FG of the heuristic, demanding around 5 Mbps less of the total bandwidth. On the other hand, for scenarios such as VNFR 1 as well as on each bifurcation of VNFR 3, where all the possible chains are disposed in a monotonic order, i.e., following and straight concatenation, both solutions yields same results in terms of demanded bandwidth.

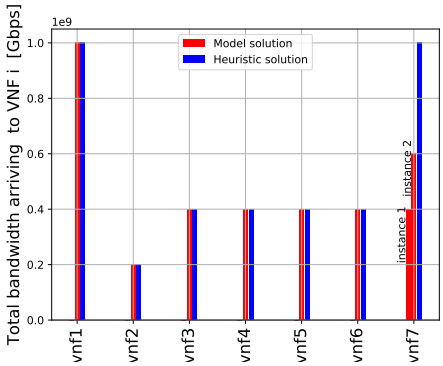
For VNFRs splitting the traffic into several bifurcations (e.g., scenarios VNFR 2 and VNFR 3), the ILP model would be able to obtain VNF-FGs with more than one instance of the same VNF for those cases in which such function must be performed by each sub-flow. This is the case of VNFR 2: for instance, where VNF 4 has to be performed by each sub-flow, our solution establishes that it must be implemented in two separately instances (one per sub-flow). Thus, the load of traffic arriving at VNF 4 is divided requiring less processing device capabilities into the physical network before the embedding process. As shown in Figure 6b, VNF 4 in our ILP solution VNF 4 is created with two instances with incoming traffic loads of 400 Mbps and 600 Mbps respectively whereas the heuristic solution maps VNF 4 to the same instance for both sub-flows processing an incoming load traffic of 1Gbps. The fact that VNFs are created in more than one instance would facilitate the subsequent embedding phase of NFV-RA.

Similarly, in VNFR 3, VNF 7 has to be located after all three sub-flows. Our ILP solution generates two instances of this function; instance 1 receives a load traffic of 400 Mbps from one sub-flow, while instance 2 receives a traffic load of 600-Mbps from two sub-flows, as shown in figure 6c.

In VNFR 1 where the VNF-FG in any combination is a monotonic graph without bifurcations, only one instance of each function is mapped. Figure 6a



(a) VNFR 1 (b) VNFR 2



(c) VNFR 3

Fig. 6: Total bandwidth arriving to VNFs

illustrates the traffic arriving at each VNF; for both solution the results were exactly the same.

Summarizing, results show that, for the simulated scenarios, our ILP model always performs better or equal than the evaluated heuristic proposal [7]. Specifically, the proposed ILP model provides better behavior when VNFRs present traffic bifurcations as it results in less total demanded bandwidth than the heuristic approach. Also, our solution creates several instances per VNF when bifurcation is present in the VNFR which would ease the subsequent embedding phase of NFV-RA as instances with less arriving bandwidth are easier to be embedded in the substrate network. The mean run time of our model considering all performed scenarios was 1.33 seg, in comparison to the heuristic with a mean run time of 0.028sec. Also, it is important to note that the objective function of our ILP was restricted here to the minimization of the total link bandwidth in the resulting VNF-FG (to be comparable with the existing heuristic). However, this objective may change depending on the operator's goals to several ones, such

as: minimization of the number of created instances, minimization of the total processing capacities, etc.

6 Conclusion and Future Work

This paper introduces a novel approach to optimally solve the SFC composition problem based on Integer Linear Programming. Evaluation results indicate that the proposed approach outperforms existing heuristic-based approaches. Specially, when bifurcation of VNFs is present in the VNFR, the proposed ILP model reduces the total incoming bandwidth and creates lighter instances of VNFs in the VNF-FG in order to facilitate the subsequent embedding phase.

Scalability issues of the proposed approach are still to be tested. Simulation scenarios were based in current IETF drafts that show only small VNFRs. A evaluation on larger scenarios to test the scalability of the approach is left for future work. Also, the extension of the ILP model to include the embedding phase of NFV-RA is an exciting branch of future research. In this way, a coordinated model including SFC composition and embedding may be created to optimally solve the two phases of NFV-RA.

Acknowledgment

This work has been funded by COLCIENCIAS, the University of Antioquia and by the Flemish fund for scientific research (FWO) and the EMD and 5GUARDS project, co-funded by imec and VLAIO.

References

1. Beck, M.T., Botero, J.F.: Coordinated allocation of service function chains. In: 2015 IEEE Global Communications Conference (GLOBECOM). pp. 1–6 (Dec 2015)
2. Beck, M.T., Botero, J.F.: Scalable and coordinated allocation of service function chains. Computer Communications pp. – (2016)
3. Elias, J., Martignon, F., Paris, S., Wang, J.: Efficient orchestration mechanisms for congestion mitigation in nfv: Models and algorithms. Services Computing, IEEE Transactions on PP(99) (2015)
4. Faizul Bari, M., Chowdhury, S., Ahmed, R., Boutaba, R.: On orchestrating virtual network functions. In: Network and Service Management (CNSM), 2015 11th International Conference on. pp. 50–56 (Nov 2015)
5. Gil-Herrera, J., Botero, J.F.: Resource allocation in nfv: A comprehensive survey. IEEE Transactions on Network and Service Management 13(3), 518–532 (Sept 2016)
6. Gurobi Optimization, I.: Gurobi optimizer reference manual (2016), http://www.gurobi.com
7. Mehraghdam, S., Keller, M., Karl, H.: Specifying and placing chains of virtual network functions. In: Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on. pp. 7–13 (Oct 2014)
8. Will, Li, H., Huang, O., Boucadair, M., Leymann, N., Qiao, F., Qiong, Pham, C., Huang, C., Zhu, J., He, P.: Service function chaining (sfc) general use cases. Internet-Draft draft-liu-sfc-use-cases-08, IETF Secretariat (September 2014)